UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Incentives for sharing heterogeneous resources in distributed systems: a participatory approach

DAVIDE VEGA D'AURELIO

UNIVERSITAT POLITÈCNICA DE CATALUNYA
Departament of Computer Architecture

Incentives for sharing heterogeneous resources in distributed systems: a participatory approach

BY
DAVIDE VEGA D'AURELIO

November 2015

ADVISORS
DR. FELIX FREITAG
DR. ROC MESEGUER

COMPUTER NETWORKS AND DISTRIBUTED SYSTEMS GROUP
DEPARTAMENT OF COMPUTER ARCHITECTURE
UNIVERSITAT POLITÈCNICA DE CATALUNYA
CASTELLDEFELS
SPAIN

Dissertation for the degree of Doctor of Philosophy
at Universitat Politècnica de Catalunya in Computer Architecture
with specialization in Communication Networks and Distributed Systems 2015

Incentives for sharing heterogeneous resources in distributed systems: a participatory approach. *November 2015*.

Davide Vega D'Aurelio
dvega@ac.upc.edu

Computer Networks and Distributed Systems Group
Universitat Politècnica de Catalunya
Esteve Terradas, 7
08860 - Castelldefels (Spain)

Alla mammma

Abstract

CONTRIBUTORY and volunteer computing ecosystems built around a community of participants need, like any other common-pool resources, an adaptive governance mechanism to guarantee the sustainability of the ecosystem. Reciprocity incentive mechanisms based on economic principles have been proved efficient solutions to regulate the resource sharing and allocation in large computing architectures, guaranteeing a direct retribution for each individual contribution even in presence of misbehaving users. However, while these mechanisms preserve the macro-equilibrium of the computational shared resources (e.g., CPU or memory), participants with fewer resources face problems competing for the attention of other members with more resources to cooperate with; making it difficult to apply such principles in practice. Additionally, active members of the community contributing in other aspects (e.g., doing administrative tasks or developing software) are not contemplated in traditional schemes although their time and effort are also part of the common-pool resource and hence, should be retributed somehow.

The aim of this thesis is to revisit some of the architectural aspects of current systems and propose a framework to govern contributory and volunteer computing ecosystems in a fairer way based on principles of participatory economics. Our main contributions in this thesis are threefold. First, we examine the mechanisms ruling the resource sharing and propose a new reciprocal incentive mechanism that measures participants' effort on sharing resources instead of their direct contribution, so it increases the collaboration opportunities of users with fewer resources in heterogeneous scenarios. Second, we propose a regulation mechanism for allocating new computational devices and distribute new resources within them, with the objective of increasing their impact in the common-pool resources when the demand of resources is supplied by the community. Third, we propose new methods to detect and analyze the social positions and roles of the community members, enabling the governance mechanism to be adapted taking into account members' effort on several tasks not considered otherwise.

The main contributions of this thesis conform a single framework that has been tested experimentally, using simulations, in a resource-sharing environment with non-strategic participants. Potentially, the mechanisms developed in this thesis will open new opportunities to apply political-economic and social ideas to the new generation of volunteer, contributory or grid computing systems; as well as other common-pool resources scenarios.

Resum

ELS sistemes de computació voluntària o contributiva construïts al voltant de comunitats de participants necessiten, com qualsevol altre *common-pool resource*, mecanismes de govern adaptatius que garanteixin la sostenibilitat de l'ecosistema. Els incentius recíprocs basats en principis econòmics han demostrat ser solucions eficients per regular la compartició i assignació de recursos en arquitectures de gran escala, garantint una retribució directa per cada contribució, inclús en presència d'usuaris maliciosos. No obstant això, mentre aquests mecanismes preserven el macro equilibri dels recursos compartits (p. ex., CPU o memòria), els participants amb menys recursos tenen problemes per competir per l'atenció dels altres membres amb més recursos quan volen cooperar amb ells; fent difícil en la pràctica aplicar aquests principis. A més a més, els membres actius de la comunitat contribuint en altres aspectes (p. ex., realitzant tasques administratives, o desenvolupant software) no es torben contemplats en els esquemes tradicions tot i que el seu temps i esforç també son part del *common-pool resource* i, per tant, haurien de ser compensats.

L'objectiu d'aquesta tesi és revisar alguns dels aspectes d'arquitectura que fan que aquestes estratègies no funcionin i proposar un *framework* per governar ecosistemes de computació voluntària o contributiva d'una manera més justa utilitzant principis de participació econòmica. Primer, examinem els mecanismes que controlen la compartició de recursos i proposem un nou mecanisme d'incentiu recíproc que mesura l'esforç dels participants mentre comparteixen recursos en comptes de la seva contribució directa, de manera que les oportunitats per cooperar incrementen pels usuaris amb menys recursos. En segon lloc, proposem un mecanisme per regular l'assignació de noves màquines de computació i recursos, amb l'objectiu de millorar el seu impacte en escenaris amb *common-pool resource* quan la demanda de recursos ha de ser subministrada col·lectivament. Tercer, proposem nous mètodes per detectar i analitzar els rols i posicions socials dels membres de la comunitat, permetent que els mecanismes de govern es puguin adaptar tenint en compte l'esforç dels participants en altres tipus de tasques prèviament no contemplades.

Les principals contribucions d'aquesta tesi formen un únic *framework* que ha estat provat experimentalment, utilitzant simulacions, en un escenari de compartició de recursos amb participants no estratègics. Potencialment, els mecanismes desenvolupats en aquesta tesi obriran noves oportunitats per aplicar idees politico-econòmiques i socials a la nova generació de sistemes de computació voluntària, cooperativa o *grid*, així com escenaris *common-pool resource*.

Preface

List of Papers

As the main author, I took responsibility for the work in all the papers. Ideas and results have been discussed with my second advisor and occasionally with the first one and, when necessary, with the other authors involved.

[P1] Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Understanding Collaboration in Volunteer Computing Systems. *Journal of Universal Computer Science*, 20(13):1738–1765, November 2014. DOI: `10.3217/jucs-020-13-1738`

Short summary This paper presents a complete study aimed to understand the impact of adopting a particular attitude to contribute with local hardware resources in a distributed shared environment.

Credits The work was planned with my second advisor. I built the simulator, gathered the data and did the data analysis. Results were discussed and written with my second advisor and the fourth author.

Assessment Q4 JCR-Science Edition (2013).

[P2] Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Effort-based incentives for resource sharing in collaborative volunteer applications. In *Proc. of IEEE 17th International Conference on Computer Supported Cooperative Work in Design*. Whistler, BC, Canada, IEEE, 2013, pages 37–42. DOI: `10.1109/CSCWD.2013.6580936`

Short summary This paper explores and analyzes a new set of reciprocity incentives, based on participatory economics principles for sharing hardware resources in a distributed shared environment.

Credits The idea was proposed by my first advisor, while the work was planned with my second advisor. I built the simulator, gathered the data and did the data analysis. Results were discussed and written with my second advisor and the fourth author.

Assessment B CORE Conference Ranking (2013).

[P3] Vega, D., Meseguer, R., Ochoa, S. F., Pino, J. A., Freitag, F., Medina, E., and Royo, D. Sharing hardware resources in heterogeneous computer-supported collaboration scenarios. *Integrated Computer-Aided Engineering*, 20(1):59–77, June 2013. DOI: 10.3233/ICA-120419

Short summary This paper analyzes the impact of several (a) well-known overlay topologies and (b) computational resources' distribution on a volunteer computing system based on the exchange of resources between peers.

Credits The work was planned with my both advisors. I built the simulator, gathered the data and did the data analysis. Results were discussed with my second advisor, the third and sixth authors. The work was written collectively by me along with the second, third, fourth and sixth authors.

Assessment Q1 JCR-Science Edition (2013).

[P4] Vega, D., Medina, E., Meseguer, R., Royo, D., and Freitag, F. A Node Placement Heuristic to Encourage Resource Sharing in Mobile Computing. In *International Conference Computational Science and Its Applications*. Santander, Spain, volume 6784. In LNCS. Springer Berlin Heidelberg, 2011, pages 540–555. ISBN: 978-3-642-21930-6. DOI: 10.1007/978-3-642-21931-3_42

Short summary This paper describes a greedy algorithm to place new computational resources in volunteer cloud computing scenarios in order to increase the cooperation between peers, by minimizing the effects of locality discovered in some overlay topologies.

Credits The idea was proposed by my first advisor, while the work was planned with my second advisor. I built the simulator, gathered the data and did the data analysis. Results were discussed and written with my second advisor and the second author.

Assessment C CORE Conference Ranking (2013).

[P5] Vega, D., Medina, E., Meseguer, R., Royo, D., Freitag, F., Ochoa, S. F., and Pino, J. A. Characterizing the effects of sharing hardware resources in mobile collaboration scenarios. In *Proc. of IEEE 15th International Conference on Computer Supported Cooperative Work in Design*. Lausanne, Switzerland, IEEE, 2011, pages 465–472. ISBN: 978-1-4577-0387-4. DOI: `10.1109/CSCWD.2011.5960114`

Short summary This paper explores the impact of resources distribution in collaborative mobile scenarios when nodes need to share their resources to perform computational jobs.

Credits The idea was proposed by my first advisor, while the work was planned with my second advisor. I built the simulator, gathered the data and did the data analysis. Results were discussed with my second advisor, the second and sixth authors. The work was written collectively by me along with the second, third and sixth authors.

Assessment B CORE Conference Ranking (2013).

[P6] Vega, D., Magnani, M., Meseguer, R., and Freitag, F. Role and position detection in networks: reloaded. In *Proc. of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. Paris, France, 2015. Forthcoming

Short summary This paper revisits the main social concepts of structural analysis, and proposes a new indirect blockmodeling framework to identify new and traditional positions and roles in more complex network models, like multi-relational networks.

Credits The work was planned and performed with the supervision of the second author during a research visit at Uppsala Universitiet. I built the library, gathered the data and did the data analysis. Results were discussed with the second author and my second advisor. The work was written by me and the second author.

Assessment Peer-review (abstract). Not ranked.

[P7] Vega, D., Magnani, M., Meseguer, R., and Freitag, F. Detection of roles and positions in multi-layer social networks. In *XXXV Sunbelt Conference of the International Network for Social Network Analysis (INSNA).*. Brighton, UK, 2015

Short summary	This paper explores and analyzes the formation of an extended relations matrix to identify positions in multi-relational graphs.
Credits	The work was planned and performed with the supervision of the second author during a research visit at Uppsala Universitiet. I built the library, gathered the data and did the data analysis. Results were discussed with the second author and my second advisor. The work was written by me and the second author.
Assessment	Peer-review. Not ranked.

[P8] Vega, D., Meseguer, R., and Freitag, F. Analysis of the Social Effort in Multiplex Participatory Networks. In *Revised Selected Papers of 11th International Conference Economics of Grids, Clouds, Systems, and Services*. Cardiff, UK,. Volume 8914. In LNCS. Springer Berlin Heidelberg, 2014, pages 67–79. ISBN: 978-3-319-14608-9. DOI: 10.1007/978-3-319-14609-6_5

Short summary	This paper analyzes the activity done by participants of a wireless community network by analyzing their interactions on two mailing-lists, while explores how this information can be used in incentive mechanisms.
Credits	The work was planned with my second advisor. I gathered the data and did the data analysis. Results were discussed and written with my second advisor.
Assessment	Peer-review. Not ranked.

[P9] Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Motivating the non-technical participation in technical communities. In *Proc. of IEEE 18th International Conference on Computer Supported Cooperative Work in Design*. Calabria, Italy, IEEE, 2015. DOI: 10.1109/CSCWD.2015.7230968

Short summary	This paper explores and analyzes a multi-layer mechanism to reward and influence the exchange of resources in one layer based on participants' actions in another one.
Credits	The work was planned with my second advisor. I built the simulator, gathered the data and did the data analysis. Results were discussed with my second advisor. The work was written collectively by me along with the second and fourth authors.
Assessment	B CORE Conference Ranking (2013).

Other papers

In addition to the above papers, I have written other work in the fields of community networks, wireless networks and distributed computing. While these publications are not part of the main contributions of the present thesis, they generated new knowledge related to the field and the working scenario. The papers are listed below.

[R1] Millán, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B., and Blondia, C. Time Series Analysis to Predict Link Quality of Wireless Community Networks. *Computer Networks*, 2015. DOI: `10.1016/j.comnet.2015.07.021`

[R2] Vega, D., Baig, R., Cerdà-Alabern, L., Medina, E., Meseguer, R., and Navarro, L. A Technological Overview of the Guifi.net Community Network. *Computer Networks*, 2015. DOI: `10.1016/j.comnet.2015.09.023`

[R3] Millán, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B., and Blondia, C. Tracking and predicting link quality in wireless community networks. In *Proc. of IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*. Larnaca, Cyprus, IEEE, 2014, pages 239–244. DOI: `10.1109/WiMOB.2014.6962177`

[R4] Vega, D., Cerdà-Alabern, L., Navarro, L., and Meseguer, R. Topology patterns of a community network: Guifi.net. In *Proc. of IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications*. Barcelona, Spain, IEEE Computer Society, 2012, pages 612–619. DOI: `10.1109/WiMOB.2012.6379139`

[R5] Vega, D., Meseguer, R., Cabrera, G., and Marquès, J. M. Exploring local service allocation in Community Networks. In *Proc. of IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*. Larnaca, Cyprus, 2014, pages 273–280. DOI: `10.1109/WiMOB.2014.6962182`

[R6] Selimi, M., Florit, J. L., Vega, D., Meseguer, R., López, E., Khan, A. M., Neumann, A., et al. Cloud-Based Extension for Community-Lab. In *IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*. Paris, France, IEEE, 2014, pages 502–505. ISBN: 978-1-4799-5610-4. DOI: `10.1109/MASCOTS.2014.73`. Demonstration

[R7] Garcia, P. E., Baig, R., Balaguer, I. V. i, Neumann, A., Aymerich, M., López, E., Vega, D., et al. Community home gateways for P2P clouds. In *Proc. of IEEE 13th*

International Conference on Peer-to-Peer Computing. Trento, Italy, IEEE, 2013, pages 1–2. ISBN: 978-1-4799-0521-8. DOI: 10.1109/P2P.2013.6688732. Poster

[R8] Aymerich, M., Baig, R., Garcia, P. E., Balaguer, I. V. i, Neumann, A., Vega, D., López, E., et al. Deploying applications with Community-Lab in wireless community networks. In *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"*. Madrid, Spain, IEEE Computer Society, 2013, pages 1–3. ISBN: 978-1-4673-5827-9. DOI: 10.1109/WoWMoM.2013.6583362. Demonstration

[R9] Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Motivating the non-technical participation in technical communities. In *XXIII Jornadas de Concurrencia y Sistemas Distribuidos"*. Málaga, Spain, 2015

[R10] Vega, D., Meseguer, R., and Navarro, L. Analysis of the web proxy service of a community network: Guifi.net. In *XXII Jornadas de Concurrencia y Sistemas Distribuidos"*. Donosti, Spain, 2014, pages 47–64

[R11] Vega, D., Meseguer, R., Freitag, F., and Navarro, L. Incentivos basados en el esfuerzo para compartir recursos en aplicaciones colaborativas. In *XXI Jornadas de Concurrencia y Sistemas Distribuidos"*. Donosti, Spain, 2013

[R12] Vega, D., Meseguer, R., and Freitag, F. Diseño e implementación de un simulador para explorar la cooperación en entornos distribuidos. In *XIX Jornadas de Concurrencia y Sistemas Distribuidos"*. La granja de San Ildefonso, Spain, 2011, pages 311–326. ISBN: 84-96737-99-0

Co-authors

Felix Freitag. Associate Professor at the Computer Architecture Department, Universitat Politècnica de Catalunya (UPC). His background includes performance evaluation of distributed systems, integration of visualization, and mechanisms for resource allocation in distributed systems. Currently, he is leading the EU funded project Clommunity.

Esunly Medina. PhD Candidate at the Computer Architecture Department, Universitat Politècnica de Catalunya (UPC). Her main research activities have been focused on ad hoc networks and routing protocols, mobile collaborative applications, context awareness, resource sharing in mobile collaboration, information visualization and distributed displays.

Roc Meseguer. Associate Professor at the Computer Architecture Department, Universitat Politècnica de Catalunya (UPC). His background include resources allocation for large-scale systems, decentralized systems applied to ambient intelligent, computer-supported cooperative work (CSCW) and learning (CSCL) and community networks based on bottom-up initiatives.

Matteo Magnani. Senior Lecturer at the Department of Information Technology, Uppsala Universitet. His main research interests span database and information management systems, specifically uncertain information management and multidimensional database queries, network science and social computing.

Sergio F. Ochoa. Associate Professor at the Computer Science Department, University of Chile. His background include mobile/ubiquitous/social computing, collaborative systems applied to urban emergencies and disaster relief efforts and software engineering. Currently, he is the Head of the Computer Science Department, University of Chile.

José A. Pino. Full Professor at the Computer Science Department, University of Chile. His background includes computer-supported cooperative work (CSCW), human computer interaction (HCI) and educational computing. Currently, he is the Head of the Collaborative Applications Research Laboratory, University of Chile.

Dolors Royo. Associate Professor at the Computer Architecture Department, Universitat Politècnica de Catalunya (UPC). His background include mobile computing, software applications and augmented reality.

Research environment

The Distributed Systems Group performs research on distributed systems in the areas of models, algorithms and software elements supporting large, complex and dynamic applications.

The ultimate motivation for our research is to empower individual people and collectives with distributed computing services, programming models and systems that mask the complexity of the myriad of interrelated software, computing and communication elements and their complex interactions in a large, diverse and changing environment.

The group is currently working on large scale and centralized community networks and community clouds, economics oriented distributed systems, particularly on resource allocation mechanisms for Grid and Peer-to-Peer, on decentralized systems applied to Ambient Networks, and on applications supporting cooperative learning (CSCL).

My PhD has been supported and funded by two European projects, CONFINE and Clommunity. Most of the contributions in this work have been evaluated and designed for these scenarios.

Community Networks Testbed for the Future Internet (CONFINE) *Integrated Project (IP) within the 7th European Framework Programme*. Universitat Politècnica de Catalunya (UPC). Code: FP7 — 288535. Oct. 2011 - Oct. 2015.

The main objective of the project was to build an experimental facility to support experimentally-driven research on Community-owned Open Local IP Networks. My participation in the project gave me access to several Wireless Community Networks — communities built around a common-pool resources with the objective to built, share and maintain network connectivity solutions —, their experimental facilities and open data. It has been extremely useful for giving context and test our theories.

A Community networking Cloud in a box (Clommunity). *Small or medium-scale focused research project (STREP)*. Universitat Politècnica de Catalunya (UPC). Code: FP7 — 317879. Jan. 2013 - June 2015.

The project aims at addressing the obstacles for communities of citizens in bootstrapping, running and expanding community-owned networks that provide community services organized as community clouds. In my research I have focused on developing regulation mechanisms based on collective participation. Additionally, most of the services organized are based on a contributory computing model in which the other contributions of this work are applicable.

Outline

This thesis consists on three parts. The first part describes the context of the research, the second part presents and analyzes the three main contributions of my work, and the last part discusses how to integrate the different proposals in a real context.

The context part aims to provide a general and rich picture of the research objectives of this thesis. The Chapter 1 introduces the research statement and the problem to be addressed. The Chapter 2 introduces each of the research questions, emphasizing how and why they relate with the main research objectives of this thesis. The Chapter 3 reviews the foundation of most of the work presented, while the Chapter 4 describes the specific methodologies, methods and tools used.

The second part contains Chapters 5, 6, 7 which describes in detail the main contributions of the present work as isolated problems that need to be addressed to fulfill our ultimate goal. These three chapters are mostly based on the publications list presented above. Thought they are connected with the same context, they should be readable independently and in a different order, provided that the reader is familiar with the corresponding methodologies, methods and tools described in Section I.

The final part consists on two chapters. Chapter 8 describes how the different mechanisms proposed in Section II can be used together in a single framework and how to generalize most of the previous results. The Chapter 9 discusses the limitations, implications and potential future directions of this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

Part I

Context

Chapter 1

Introduction

Advances in computing and networking technologies improve everyday. The cost reduction of powerful computing devices — including mobile devices — and network connections make them accessible to most people, opening new opportunities for developing distributed architectures aimed to share computational resources.

Different computing paradigms appeared to model *how* resources are shared and/or aggregated and *what* roles are assigned to the users of the system. In this thesis we focus our attention in computing architectures for cooperative systems, in which the pool of resources is owned and shared by the users themselves — who will act simultaneously as providers and consumers —, like volunteer and contributory computing.

In *volunteer computing* [1, 2] architectures, for example, users combine dedicated and non-dedicated resources to perform complex computational tasks split in smaller jobs run independently on each device. In *contributory computing* [3], however, jobs run on top of long-lived decentralized services distributed within the users' computational resources. Despite the differences on how jobs are assigned to each device, in both models the participating users are engaged in a cooperative [1] process.

In both architectures, from the point of view of users, running tasks or deploying services has no apparent cost. The extra resources needed are temporally borrowed from other users transparently by the middleware software, giving them the false sensation of unlimited access to an unbounded amount of resources. In reality, there is a competitive component as the number of aggregated resources is actually limited by the total amount contributed by all users and the dynamics of the system. This cooperative model — where a set of non-excludable goods are used in a competitive environment — has been named as a *common-pool resource*.

Hardin [5] argued that the unrestricted access to a common-pool resource leads to a saturation point, where the individuals demand of resources is higher than the overall provision. The same problem has been found in other distributed schemas, like Peer-to-Peer file-sharing networks, where some users — called *free-riders* — intend to improve their bandwidth usage by downloading content from the network without contributing with their own content. This problem, formally known as the *tragedy of the commons* has been subject of research in the fields of economics, sociology and computer science.

Contributory-based incentives have been proposed as a regulation mechanism for common-pool resources in decentralized computing architectures. The general idea behind these mechanisms is to distribute the computational resources (e.g., CPU or memory) according to the users past contributions. They have been proved good solutions to maximize the fairness (in terms of resources consumption and contribution) for each user locally. Although these mechanisms guarantee a sustainable computing system, in practice they punish users with fewer resources in highly heterogeneous scenarios.

To briefly recall the problem, consider the Figure 1.1 matrix of uploaded traffic between each pair of peers — averaged over all runs — in a Peer-to-Peer experiment performed by Legout et.al. [84]. Darker squares represent more data (bytes). Peers 1 to 13 have a lower upload limit, peers 14 to 27 have a medium one and peers 28 to 40 have a higher upload limit. In the experiment, the common-pool resource — bandwidth — is regulated by a contributory

[1] Cooperative in the context of this thesis means that the “interaction is characterized by positive goal interdependence with individual accountability” [4]

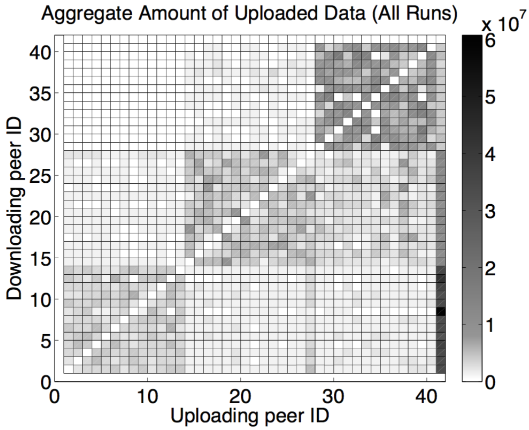


Figure 1.1: Aggregate amount of uploaded data in a Peer-to-Peer experiment

mechanism, that enforces each node to download data proportionally to how much it uploads. While it prevents the collapse of the Peer-to-Peer system, we can observe that, as a consequence, nodes with larger resources tend to cooperate only among them, because otherwise they will see their download ratio limited. Therefore, nodes with fewer resources are constrained to upload and download data only among them too. It results in a cooperative divided scenario, that punishes particularly nodes with fewer resources.

Problem statement

In computational systems with heterogeneous resources, the cooperation clustering problem might prevent punished nodes from getting enough resources to start their own jobs. Therefore, users will not see any reason to remain in the ecosystem, eventually causing a lack of resources and leading it to its saturation point.

The aim of this thesis is to revisit some of the architectural aspects of current systems and propose a framework to govern contributory and volunteer computing ecosystems in a fairer way based on principles of participatory economics. *Participatory economics* [7] is an economic system based on decision-making, where the rights — the ability of decide of some individual — is proportional to their actions — how much he had made for the community. Therefore, the economic system is based on policies that *compensate the effort or sacrifice* rather than absolute contributions.

Proposed solution

Our working hypothesis is that *effort-based incentives* will achieve higher cooperation than contributory-based incentives and will be more inclusive with those users with fewer resources, provided that the incentives are enforced by some external entity — the community of practitioners.

As we will demonstrate, effort-based incentives will normalize the cooperation opportunities of each participant, canceling the negative effects of the heterogeneity. While these strategies are strictly equitable, it is true that nodes with more resources could increase their individual outcome — in detrimental of the overall welfare — by using traditional strategies instead. Therefore, our mechanisms require enforcing its adoption.

Ostrom [6] observed that in many common-pool resources (e.g., farming fields) this task is managed successfully as a collective action, instead of lying on some external ruler. Following the same principle, in this thesis we explore the idea of using the community of user of the cooperative applications to enforce our resource sharing mechanism. This solution, however, requires an active participation of users — who will need to devote part of their time and effort to this task. Therefore, our framework also includes a mechanism to encourage users' participation in supporting tasks by compensating them with extra resources in the common pool.

Chapter 2

Research statements

The purpose of this research is to guide system designers for cooperative applications and members of common-pool resources on developing government mechanisms more inclusive, by providing fairer collaboration ratios among their members. To that end, in this thesis we propose and evaluate a new framework for regulating resource sharing inside cooperative applications' communities.

To accomplish the above objective this research focuses on designing and understanding new incentives capable of measure, compare and use the effort (e.g., time, computational resources) devoted by the members of communities built around cooperative applications' to improve the welfare of the community, and use them to increase the overall collaboration.

While the specific contributions of this thesis are presented in detail in chapters 5, 6 7 and 8; in this chapter the main research questions and results are introduced.

Encouraging resource-sharing efforts

In cooperative applications, substituting traditional reciprocity-based incentives for others based on participatory economic principles will increase the collaboration opportunities of the poorest participants (in terms of resources)? Assuming that this is true, *Does the new incentive guarantee a larger percentage of successful jobs?*

As we have introduced, the free and unrestricted access to common resources might lead to an unsustainable situation, where the overall demand for computational resources exceeds the current provision. It is a consequence of the selfish behaviour of most users, who believing that their increment of demand would not be noticed they try to increase their profit by consuming more resources than other users.

Reciprocity-based incentives [8] has been proved good solutions to regulate how to acces shared resources. By construction, in reciprocity-based incentives the consumption of resources is proportional to some evaluation function, measured directly or indirectly by peers. Almost all reciprocal incentives are also contributory-based, meaning that the evaluation function is proportional to the amount of resources the requesting user had contributed previously with. As an example, in bitTorrent file-sharing system [9] the amount of downloading traffic allowed to an user is, in average, equal to its upload traffic multiplied by a constant.

One key advantage of contributory systems [1] is that encourage users' participation by rewarding their good will, instead of punishing those not participating. Following our example of bitTorrent, the only way users can get more resources from the network is by increasing their contribution and being connected with users with higher cooperation willingness. As a result, users are rearranged by sharing capabilities and willingness into clusters of similarity. Therefore, users with fewer resources are in unfavorably positions when they compete for possible resources, although they are more willing to cooperate than other users with larger amounts of resources.

We believe that if users' reciprocity would be measured in relative terms rather

[1] In this thesis we use the term contributory-based incentive as a synonymous of "reciprocal and contributory incentives"

than in absolute values, then any participant will have the same opportunities to cooperate. Chapter 5 introduces a series of reciprocity-based incentives inspired by the same principles of participatory economics, which uses indirect and direct relative evaluation functions. We show that these mechanisms lead to scenarios where poorest participants (in terms of resources) achieve higher cooperation ratios than in traditional incentives. It has a positive effect on the sharing architecture, increasing the overall number of jobs fully satisfied.

However, as the new mechanisms have been designed to encourage the participation of a small fraction of the population, it could rise concerns in other users with larger amount of resources.

Is this incentive observed fair for nodes with larger amount of resources? What are the practical implementation issues?

Answering the first question required an extensive analysis of the resources distribution, which shown that our mechanism not only increases the overall cooperation willingness; but also increases the utilitarism of the computational resources.

The second question has been addressed by analyzing the robustness of the mechanism in the presence of liers nodes — those misreporting their maximum sharing capacities — and uncertainty in the evaluation function.

The main results presented in Chapter 5 related with this contribution were originally reported in [P1, P2].

Provisioning resources in scarcity scenarios

In higher heterogeneous architectures for cooperative applications, under which conditions devices with fewer resources are able to cooperate with others? *What additional resources are needed to encourage collaboration among them? Is there an ideal ratio between nodes with few and many resources that maximizes the overall cooperation willingness?*

Lack of computational resources in highly dense overlay topologies might slow down the growth of the community of users, making hard for users to get enough resources for running complex computational jobs without bypass some of its peers. In a direct reciprocity scenario, for example, two connected users are only allowed to share with each other a limited amount of resources. As the average degree of the nodes involved in the collaboration process increases, they need to reduce the maximum of resources given to each other in order to avoid the tragedy of the commons.

One possible solution that we explore is to encourage the community members to borrow or deploy new computer devices, managed individually or collectively, to supply the demand. Then, various questions arise when we try to analyze the cooperation support provided by several overlay network topologies involving heterogeneous hardware devices. In particular, it is interesting to study the effect of introducing devices with higher computational capabilities in a scenario without much resources per node, which would probably change the network behaviors; turning the system into a high heterogeneous scenario with two clusters of nodes.

The analysis performed in Chapter 6 and [P5] reveals that in direct reciprocity scenarios, introducing external resources improves the level of cooperation among nodes with fewer resources, provided that the ratio between extra computational devices and extra resources is maintained. As an example, introducing only one extra device with huge amount of resources has a minimum impact on the collaboration process if the exchange of resources happens between nodes directly connected.

Various related works [10, 11, 8, 12] show that network topology characteristics play a role in the collaboration, and that it can also encourage and promote cooperation if certain conditions are present. Thus, we not only expect that introducing devices with more resources in the network will promote collaboration, but we also believe that the collaboration level is highly dependent on the network topology and on how resources are distributed among nodes. Consequently, new research questions arise:

Are the topologies used to support collaboration in various fields (e.g. game theory, neural networks, file sharing) applicable to resource sharing on a heterogeneous collaboration scenario? If the answer is yes, then *Is the nodes' placement strategy a variable that can be used to improve nodes' cooperation?*

These research questions are not relevant in some architectures like cloud or grid computing, where the cooperative applications have no visibility or control on the network topology. Therefore, nodes cannot take particular actions (e.g. increasing the network degree), to improve the collaboration among them. Instead, in cooperative applications — where the exchange of resources is controlled by the overlay topology — who is allowed to share (and compete) with can be controlled easily. Furthermore, community members of resource-pool resource might enforce them by social pressure.

The answers to these last questions are the central topic of Chapter 6, which is based on the research results presented in [P3, P4].

Effort-based incentives for cooperative applications

How users' participation on supporting activities can influence the sharing process? *What are the components needed? Is it possible to influence supporting activities by users' participation in the sharing process at the same time?*

We have seen how effort-based mechanisms encourage the cooperation in scenarios with higher heterogeneity of computational resources. In the absence of nodes with large amount of resources, we have provided a placement algorithm to supply this demand with collective computer devices. Both mechanisms are indifferent to selfish or misbehaving nodes as they assume the existence of a community of users supporting and governing the shared infrastructure.

In the first part of Chapter 7 and [P8] we have studied the participation of members in a common-pool resource aimed to build and share network connectivity, and have observed that only a very small fraction of the population are willing to overtake these tasks. Thus, we focus our attention on how the community members can be encouraged to participate in other aspects of the ecosystem besides the main activity — in this case, sharing and building network infrastructures.

We believe that a new regulation mechanism, able to provide some reward based on how much users participate in the supporting and governing tasks, can increase the number of users willing to participate. In Chapter 8 and [P9] we focus our attention on designing and evaluating the new regulation mechanism, under the assumption that there exists some externality — maybe a social process, or another mechanism — scoring each community member based on its participation in the supporting activities.

The framework uses this score to modify the information exchanged among users, allowing users with higher scores to announce smaller values of resources own, increasing then the ratio of shared and own resources. As the social-effort scores are managed privately by the economic framework — and are not shared with any user in the system — users cannot distinguish in which activity their neighbours are really participating. This design decision has the advantage of preventing strategic users from getting undeserved advantages or to concentrate their efforts on just one activity. The Chapter 8 focuses on designing and evaluating this new mechanism.

Is it possible to generalize the framework for architectures with multiple resources or supporting activities?

Based on the previous results, we generalized our framework to work with multiple resources and supporting activities, so that the amount of resources a given user gets from its neighbours is based on its participation in the overall community, rather than a particular activity. We also show how our generalization allows system designers to weight the impact of every activity individually, enabling each common-pool resource community to design their own governing policies.

These design decisions follow the general principles of participatory economics, in which users' reward is a measure of the effort they devote to a particular activity, as well as how are their participation complexes and interests balanced.

Social positions and roles detection in multiplex graphs

Can the participation of users in supporting activities be detected analytically? Is it possible to discriminate between participative actions useful for the community and those not?

One of the main difficulties on scoring users based on their participation in supporting activities is to define the problem properly: what we understand by *participation*? And, on which *supporting activities* we want to measure it? An indirect approach would be to analyze what these supporting activities have in common, independently of the kind of participation, and find an indirect measure of the level of participation — social effort.

Previous experiences on open software communities [13] and community networks [P8] teaches us that nowadays the technology is present in most of the decisions and interactions that occur in the context of these communities of common-pool resources. As an example, guifi.net uses generic mailing-lists to coordinate collective actions and specific mailing-lists to discuss regional problems in their infrastructure. The use of online social platforms or participatory forums is twofold: (1) the digitalization of the interactions can be stored and evaluated easily, and (2) the social process driven by the community can be used to assest the quality of the information [14] and hence, its utility.

Therefore, we mapped the *social effort* as the influence of users in the overall ecosystem, measured through their interactions with other users and the utility of those interactions. According to our model, we modeled these interactions as a multi-layer graph, where vertices represent users, weighed edges represent some interaction and each layer represent one of the possible venue where this interaction might occur (e.g., one mailing-list, a face-to-face meetings). Our initial analysis of the social structure, based on community detection [P8] revealed little information about the influence of users across multiple social venues.

Is it possible to detect social roles and positions in complex graph structures — like multi-layer or multi-relational graphs — based on multiplex distance measures? *Can any similarity measure being detected?*

Roles and *positions* analysis [15], instead, have been used previously to group users according their influence in complex graph structures [16]. However, traditional methods are limited to positions (or roles) based on simple graph measures, which easily can ignore positions related with multiple layers. *In Chapter 7 and the publications [P6, P7] we develop a new framework for the detection of social roles and positions for complex graph structures.* This framework computes the similarity and dissimilarity among actors using comparisons between actors and sets of actors instead of just using pairwise comparisons, which allows us to include multi-layer network measures in the analysis.

Using our developed framework, we introduce some new similarity measures that might be used to find roles in multi-relational graphs, and discuss how to measure objectively their level of uncertainty.

Chapter 3

Foundations

The work presented in this thesis is supported by background concepts and theories from different research fields. In this chapter we revisit all of them. We first define the computer architectures for cooperative applications, emphasizing the role of community members in the supporting activities. Then, we introduce the need of implementing regulation mechanisms in distributed architectures to avoid the tragedy of the commons while encouraging the participation of users. Finally, we review some basic concepts about the analysis of simple and complex networks, focusing on the different network models and properties used or analyzed in this thesis.

This chapters aims to provide a basic background on these topics for non-experts in information systems, computational analysis or social network analysis. Experts might skip this chapter and use it as a reference guide, if needed, to understand other concepts presented on Part II.

3.1 Computational network architectures for cooperative applications

The first *computational network architecture* known for sharing resources dates back to the later 90's, when the computational capabilities of personal computers became insufficient for the research ongoing in academic institutions. As a solution, in 1994 Foster et.al. [17] created the first network-based computing architecture, based on the aggregation of distributed resources to perform a common tasks, later known as *grid computing*.

The original work on grid architectures focused on creating protocols for the task execution, synchronizations and resource management [18]. Access and rights over the use of the infrastructure were controlled off-line, by the responsibles of the project and later on through a centralized web platform. The single restriction imposed to the use of the shared infrastructure was due the technological impediments to perform large batch tasks.

A similar idea of aggregate computational resources was popularized some years later by telecommunication companies under the name of *cloud computing*. Clouds were created with the purpose of provide computational costly services — not only physical resources — to external users as an integrated service though the network. The governing model was then monetary, as users pay as they consume more services and infrastructure.

Two main differences distinguish cloud and grids. Firstly, from the technological standpoint cloud infrastructures reallocate the tasks and jobs into virtual environments inside computational devices hosted by the cloud provider [19], allowing them to maximize the utility of hardware resources. In grid computing, instead, the jobs are executed on top of the operating system. This was one of the main reason of its success and economical viability. The second difference is the change of the sharing model implemented, as in cloud computing the providers and consumers are two different agents. In the cloud computing model, users are borrowing services at a market cost, that are held and managed by someone externally; while in grid computing users are at the same time consumers and producers.

3.1.1 Volunteer and contributory computing

Volunteer and contributory computing are two different computational models for sharing resources and services in a Peer-to-Peer (P2P) model. P2P models [20] are computer distributed architectures traditionally used for sharing files between geographically disperse users over the network. They are known for enabling the direct exchange of information between end-users, creating a new model of computation where users act at the same time as servers — providing content — and clients — consuming content from other users.

Two of the key challenges in P2P systems have been the self-organization of the content and its indexing over a large amount of interconnected computer devices. Overlay networks are an important component to accomplish these goals. They can be viewed as a virtual topology built on top of the real network, able to routing messages among nodes efficiently. Overlay networks have been used since then in multiple distributed systems like volunteer and contributory architectures.

Volunteer and contributory computing use then similar network architectures, even if their computational model is different. *Volunteer computing* [1, 2] systems are intended for combining dedicated and non-dedicated resources to perform complex computational tasks. These tasks, are organized into small jobs — in a similar way as how grid computing divides its tasks into batches — which are then executed independently in multiple peers' devices. *Contributory computing* [3] instead, divides splits and executes long-lived decentralized services (e.g., lookup, authentication) and provides an unified framework to access the platform and run complex tasks on top of them.

In this thesis we will embrace both concepts, *volunteer computing* and *contributory computing* under the definition of **cooperative applications** because both systems are aimed to share computational resources using a similar approach — executing small jobs on P2P network architectures —, although with different philosophy. In both systems tasks are allocated — in long or short term — and executed over a set of peers connected though an overlay network at no apparent cost, giving users a false sensation of unlimited access to an unbounded amount of resources.

The cooperative model of volunteer and distributed computing is very similar to the economical model of common-pool resource scenarios, where consumers have unlimited access to a set of non-excludable goods — computational resources in our case — in a competitive environment.

3.1.2 Beyond the computer architecture: communities

Nowadays, most open-source software (OSS) [21] projects have a growing community of users supporting them. In most cases this is due the lack of official support, which highlights the need for some forum where users can share experiences or discuss problems [13]. Additionally, new and easily usable collaborative and management tools, like distributed version control and web-based integration systems, are attracting new users with less experience in OSS that would not participate otherwise. This ecosystem is very similar to the communities of practice described by Wenger [22, 23], where the social aim of learning drives the growth of social ties among members interested in a common shared domain of interest.

Some OSS communities have a regulation system to control the quality of the software contributed by its members. While some members feel that, in this context, the mechanism encourages good practices and coding, others feel that the mechanism is more exclusive than inclusive as it creates a barrier for users' participation [24].

We believe that in cooperative software applications — like volunteer and contributory computing — members of the communities of practice can have a similar role as in open-source software projects governing and ruling the collaboration process between users implicitly. This scenario will have much more similitudes with the governing mechanisms observed by Ostrom [6] and therefore, it will open new opportunities for implementing more inclusive and fair regulation mechanisms in heterogeneous resource scenarios.

Community networks

Community networks are growing fast as a sustainable model for self-provisioned computer networking infrastructures [25], alternative to other service offerings. This has been accelerated by the reduction of the costs of WiFi and optic networking equipment, combined with the growing popularity of wireless devices, and the lower complexity of network setup. In recent years, a plethora of non-profit initiatives have flourished to create community networks providing, among other services, Internet access. A few examples are guifi.net [26] and FunkFeuer [27].

A characteristic of these initiatives is that the network topology grows organically, without a planned deployment or any consideration other than connecting devices from new participants or locations by linking them to an existing one or improving the network service. Typically, the deployment and management tasks are performed by the community network members, mostly volunteers [28].

Beyond Internet access provision, the community networks' physical infrastructure is sometimes used by some members to provide applications (e.g. web servers, monitoring systems). As a natural evolution, some community networks are looking for ways to implement higher level services [29], which would require mechanisms to regulate and normalize how their members interact with the computational resources [P3]. The feasibility of implementing such contributory systems is highly dependent on the network participants' ability to rank and evaluate members' participation.

Community networks are, therefore, an example of common-pool resource [30]. They have a self-provisioned common good shared without barriers among the participants, while managed and governed by community agreements. We used guifi.net and other community networks for gathering and analyzing communities behaviour in a common-pool resource context.

Contributory clouds

During the last decade, a whole series of new computer network architectures have flourished as a cloud solutions for sharing resources incorporating some principles of the contributory model. One common aspect of these new architectures is their emphasis on the community aspect.

As an example, Cloud@Home [31, 32, 33] project proposes an architecture to combine cloud and volunteer computing paradigms. Their model proposes a federated cloud architecture, composed by a set of autonomous clouds built with user-contributed resources and with the capability of consuming and lending resources from and to commercial clouds if needed. Therefore, communities are defined in their context as a computational and architectural entity — each of the autonomous clouds —, which have the control over the resources/services placed on its devices according to a credit mechanism. The main focus of the project is, however, to achieve transparent interoperability among the different autonomous systems.

A similar paradigm, named Community Cloud Computing (C3) [34, 35], appeared around the same time as a cloud infrastructure "with nodes potentially fulfilling all roles, *consumer*, *producer*, and most importantly *coordinator*". In contrast with cloud@Home, C3 architectures are based on a distributed model where shared resources and server functionalities are provided by end-users. This model can empower small communities of users to deploy their own community cloud, based in a commons model. Communities are, in the context of C3 architectures, a social and economic entity which can actively interact with the shared infrastructure and services.

In practice, and as far as we know, only the Clommunity [36] project implemented some of these ideas. As we said before, the main objective of the project was running and expanding community-owned networks that provide community services organized as community clouds.

While the work developed in this thesis can be viewed as a first step towards incentive mechanisms for community cloud computing architectures, the volunteer and contributory computing models to which it is addressed are architecturally different. Firstly, our proposal is aimed to share computational resources instead of virtualized machines or containers, like in cloud computing architectures. Therefore, other aspects characteristics of cloud systems (e.g., VM migration) are not evaluated in this work. In second term, the sharing model proposed assumed that users participate with cooperative intentions — sharing resources on short-term — instead of being collaborative — with long-term common goals. This difference also holds for contributory architectures as the computational model is based on resources, despite the middleware manages services.

3.2 Regulation mechanisms

Regulation mechanisms are a fundamental part of any distributed computing architecture, despite if it implements or not a descentralized model, or if the governing model is collective or individualist. The human being is selfish by nature, and hence it needs bounds (e.g., laws and rules) to enforce its behaviour in a society. In this section we introduce the need of regulation in cooperative applications and review how most of the reciprocity-based incentives work.

3.2.1 The tragedy of the commons

Two important characteristics differentiate common goods from other type of resources: *rivalrousness* and *non-excludability*. In economics, rivalrousness is a property of some goods which states that its consumption — temporal or definitive — by some individual, precludes its consumption from others. A good is non-excludable if it is not possible to prevent individuals have not paid for it to access it. The main problem is that individuals — potential consumers of the common pool — are aware of the first property (rivalrousness) and try to take advantage from the second one (non-excludability) without noticing that, in fact, the common good might not be infinite.

While the access to common goods is unrestricted, users might be tempted to consume as many resources as they can, following the false idea that it will not impact negatively to the overall system. Given the rivalrousness of the system, other users might join them on this strategy until a point of saturation, where the individuals demand of goods is higher than the overall provision. This problem was described as the *tragedy of the commons* by the ecologist Garrett Hardin [5].

Most of the distributed resource sharing systems presented early on for sharing resources can be considered a common good. As an example, users in Peer-to-Peer file-sharing systems compete by the bandwidth capacity, which is a rivalrousness good — the bandwidth consumed to a user downloading a file cannot be consumed by anyone else — and non-excludable — once the users are logged into the Peer-to-Peer system there is not excludability on bandwidth consumption.

To address the problem, most systems implement explicit governing mechanisms with a two-fold objective: (1) encourage users to contribute with more resources, and hence increasing the limit of consumption and (2) regulate how much each user can consume to avoid the tragedy of the commons.

3.2.2 Prisoners' Dilemma framework

The prisoner's dilemma (PD) [37] is a well known framework based on *game theory* [38] to study the cooperation process between two individuals with opposed goals but common welfare. The game consists on two players who must decide individually — and without the possibility to negotiate — to cooperate (C) or defeat (D) the other co-player. The payoffs for the two actions are shown in Table 3.1.

Table 3.1: Payoff matrix of Prisoner's Game

Player decision	**Co-player Cooperate**	**Co-player Deflection**
Cooperate	(a, a)	(c, b)
Defection	(b, c)	$(\varepsilon, \varepsilon) \to 0$

The relations between different possible payoffs follow the rule $b > a > \varepsilon \to 0 > c$, which immediately poses the dilemma: if cooperation is costly for the individuals and it benefits only the interaction partners, then Darwinian selection should favor non-cooperating defectors and eliminate the cooperators. This leads to a highly inefficient outcome compared to the results obtained by two cooperators.

We have chosen this game because it captures the relation between the common-pool and the local decisions taken by users in our scenario. The cooperation choice (C) represents the willingness of sharing computational resources with other users, which is rewarded in a competitive way: if the co-player(s) choose to share resources too the economy of the common pool will be sustainable and both are rewarded with higher payoffs. Each other choice based on one or more co-players not contributing will lead to an unsustainable system without many resources.

3.2.3 Contributory-based incentives

The *contributory-based incentives* are defined by Rahman et.al. [39] as sharing strategies that only consider the past contributions performed by the co-player, when a player has to decide whether to cooperate or defeat with him.

In the Prisoners' Dilemma game, a well-known strategy to maximize the payoff when using this approach is the *Tit-For-Tat (TFT)* strategy proposed by Axelrod et.al [40]. Players using this strategy always start their interaction with another co-player cooperating with him and, in the following interactions they replicate the answer of the co-player in the previous round. The success of BitTorrent protocol [41] is an example of a real application using Tit-For-Tat as a core strategy. Resulting from that decision, nodes contributing more resources should receive, in return, a better service than those that contribute less [9, 42].

However, the study and analysis in [43] shows that forgiveness is a necessary condition to achieve cooperation. Consider the example of a player that, for any reason, defeats his co-player one single rounds. If both players are using a TFT strategy, the next round the co-player will choose to defeat, while the previous defeating player will cooperate. From then on, players will switch roles each round between cooperation and defection. Even worst, if by mistake both players defeat the same round each other, then there is no turn back to a cooperation situation unless someone makes another mistake. In cooperative real applications forgiveness is implemented in order to speed-up cooperation reconstruction after single defections. Most BitTorrent clients use opportunistic unchoking to compensate nodes' mutual defection caused when one peer's chocking policy replaces one link [44].

3.3 The study of complex networks

It has been shown there are some topological patterns and graph characteristics promoting cooperation among the nodes in real world networks [45, 11]. This section provides a brief description of these models and the main characteristics evaluated during the evaluation of our contributions. For a more complete description of the content described in this section we refer the reader to [46, 15].

3.3.1 Network models

Different network models have been used during the evaluation of this thesis to describe social structures (e.g., interactions among members of a community) and communication networks (e.g., the overlay topologies used by some cooperative applications). Next, there is a summary of the different models used.

- **Torus**: An n-dimensional torus is defined as the Cartesian product of n rings. Symmetric torus networks are built from rings of the same length. A constraint is that the number of nodes must be D^n, where D is the number of nodes in the ring.

  Under uniform traffic circumstances, torus topologies have the advantage of providing a balanced use of the network resources. A symmetric torus improves the mesh by connecting the head node with the tail node in each row and column. Thus this topology eliminates the edge effect.

- **Erdős–Rényi (ER) random graph [47]**: is a static random model where each link has an independent probability of being formed p. Therefore, in the or $G(n,p)$ model the probability of any node n having k links is binomial:

  $$P(d(i) = k) = \binom{n-1}{k} p^k (1-p)^{n-k-1} \tag{3.1}$$

  ER random graphs are considered the most simple random probabilistic networks and are of great interest for comparative analysis.

- **Waxman random graph**: The Waxman's probability model for interconnecting nodes [48] was used to build a random network topology based on ER model, constrain to some 2D lattice L^2. Edges are introduced between pairs of nodes i, j with a probability that depends on the distance between them:

  $$P(\{i,j\}) = \alpha e^{\frac{-d(i,j)}{\beta L}} \tag{3.2}$$

  where $d(i,j)$ is the Euclidean distance from node i to j; α is the probability of edges between any vertex in the graph and controls the average degree of the network, β is the ratio between long and short edges and L is the maximum distance between vertices.

  As nodes are distributed uniformly random along the surface, $d(u,v)$ is our random variable $[0,L]$. Thus, the Waxman model generates networks with lower variability of nodes degree and smaller diameter size than other Internet topology generators [48]. The model also has an exponential clustering coefficient distribution, independent of the network size which is representative of most random graphs.

- **Watts–Strogatz (WS) [49]** is a random graph model that produces network topologies with small-world properties by rewiring edges in a regular ring lattice. Compared with the ER model, the Watts–Strogatz presents a higher clustering coefficient and a power-law distribution of nodes degree as opposed as the Poison approximation shown by ER model. Our interest in this model is to compare how the clustering of certain nodes might influence the cooperation process.

- **Barabási–Albert (BA) model [50]**: is a growing random network based on preferential attachment, which generates network topologies with a small number of nodes acting as hubs (nodes with larger degree), while most nodes have a low degree.

  According to the incremental growth of the nodes' power degree, the network starts with an initial set of m_0 connected nodes. Each new node is connected to $m \leq m_0$ existing nodes with a probability that is proportional to the number of links that the existing nodes already have. Hence, the probability of interconnecting a new node i with node j belonging to the network — the $\{i, j\}$ edge probability — is given by:

$$P(\{i,j\}) = \frac{d(j)}{\sum_{j \in V} d(j)} \tag{3.3}$$

  where $d(j)$ is the current degree of node j to which node i would be attached, V is the set of nodes which joined the network. The lower term is the sum of out-degrees of nodes that previously joined the network.

3.3.2 Properties of networks

Table 3.2 summarizes the main characteristics of the network models that we discuss during the analysis and evaluation of our mechanisms: degree distribution, average path length scalability, and clustering coefficient scalability.

Table 3.2: Properties of network models

Network models	Degree distribution	ASPL scalability	Clustering
Torus	constant	$\mathcal{O}(N)$	$\mathcal{O}(1)$
Erdős–Rényi	low variability	$\mathcal{O}(\log logN)$	p
Waxman	low variability	$\mathcal{O}(\log N)$	exponential
Watts–Strogatz	$P(\lambda, k)$ limit	$\mathcal{O}(N)$	independent
Barabási–Albert	$P(k) \sim k^{-3}$	$\mathcal{O}(\log N)$	$C \sim N^{-0.75}$

- **Degree distribution**. The degree of a node in a graph is the number of arcs edges to other nodes. The degree distribution is the probability distribution of these degrees on the whole topology.

- **Average shortest path length**. The average shortest path length (SPL) is the average number of hops between each pair of graph nodes using their minimum path. The ASPL scalability shows dependency between the SPL and the network size.

- **Clustering coefficient**. The local clustering coefficient of a node in a graph is the proportion of vertices of the node to the number of all possible vertices. The clustering coefficient distribution is the probability of these coefficients in the whole network.

3.3.3 Structural analysis

Structural analysis on networks intends to capture and interpret how nodes are related according to the network topology. When applied to social networks, structural analysis is able to identify key actors or groups of actors whose connectivity influences in somehow the dynamics of the system. Three typical ways of grouping actors based on their connections consist in identifying *communities*, *positions* and *roles*. While related, these are three distinct types of groups and they typically require distinct algorithmic treatments.

To briefly recall the difference between community, position and role, we use Padgett's social network representing business relationships among Florentine

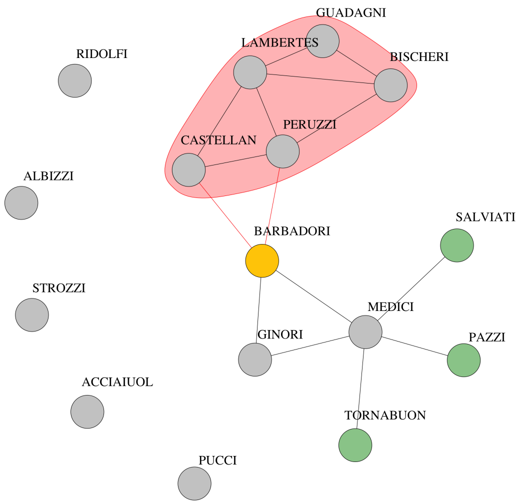


Figure 3.1: Padgett's business family network

families during Renaissance [51] as an example (see Figure 3.1).

As we can observe, a community indicates a cohese group of actors, with many connections inside the group and fewer relationships with other actors outside it. For example, the five nodes on the top of the figure form a community (red area). Roles and positions, instead, focus on the interchangeability of the actors, and do not require any internal connectivity. In its simplest form, a position is defined as a group of actors who are similarly connected to other actors in the network (green nodes). In our example, the families *Salvati*, *Pazzi* and *Tornabuon* are in the same position, because they are all connected to the *Medici* family and to no other family. Roles, instead, refer to actors with similar patterns of connectivity, independently of the specific actors to whom they are connected. In our running example, the *Barbadori* family (yellow node) has the role of connecting two otherwise disjoint parts of the network. From this point of view, it does not matter who exactly is connected to them: if the *Barbadori* family were connected to *Salviati* instead of *Ginori*, it would still play the same role in the network, but from a different position.

The methods and tools used to explore these network structures are discussed in detail in Section 4.

3.3.4 Multiplex network models

In most real systems, actors interact with each other in complicated patterns that cannot be captured by simple — or monoplex — network models. We can view such systems as a set of sub-structures — layers — where their components — actors — are present in one or more of them, and interactions occur within components of the same or different sub-system.

Multiplex — or multilayer — networks are a mathematical graph structure to describe such systems. More formally, Kilevä et.al. [52] defined a multiplex network as a quadruplet $M = (V_M, E_M, V, L)$, where V_M is defined as the set of possible nodes present in each layer L and E_M is defined as a set of pairs of possible combinations of nodes and layers. That is, $E_M \subseteq V_M x V_M$ where $V_M = V x L_1 x \cdots x L_d$ and d represents the total number of layers.

Figure 3.2 is a representation of a multilayer network with three layers ($d = 3$) and six vertices ($V = 6$, $V_M = 9$). We colored black the edges between vertices of the same layer — often called intra-edges — and in dashed red lines the edges between vertices in a different layer — called inter-edges.

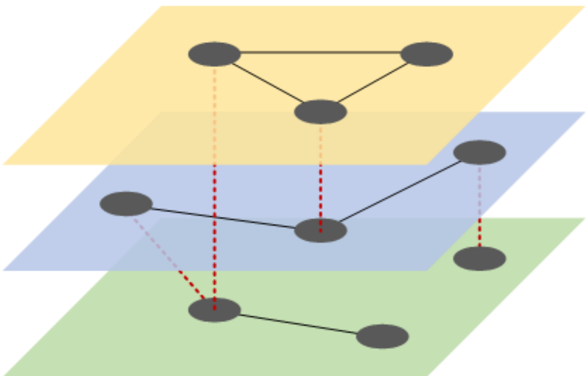


Figure 3.2: Multiplex network

As we said, the multiplex networks are good structures to represent complex relations, but they can be easily used to represent more simple graphs, like multi-relational graphs (in this case, each layer will represent one type of relation and all vertex i will be present on each layer).

We used multiplex graphs in this thesis to relate the activities performed by users in the common-pool resource with their social activities in multiple social forums.

Chapter 4

Methodologies and methods

This chapter is an accounting of the research methodologies, methods and tools used during the realization of the present thesis. They are discussed in relation to the design science model proposed by Peffers et.al. [53] for conducting research in Information Systems. The main objective is to give both, an assessment of the quality of the research and a description of those methods and tools that the reader might not be familiar with.

The first part of the chapter describes the methodologies used to analyze, model and evaluate each one of the contributions. The second part focuses on describing the main artifacts used. Some of the artifacts are well established methods (e.g., the community detection methods), while others are created during the realization of the present work (e.g., a non-cooperative game simulator based on discrete events). There are other artifacts not explained in this chapter, which are considered a — or part of a — contribution and, therefore, are explained in the corresponding chapter.

4.1 Methodologies

4.1.1 Data analysis

Wireless community networks — and particularly guifi.net — have been an important source of information to understand the interactions among community members in a common-pool resource [P8]. The analysis of such scenarios required gathering, modeling and analyzing large sets of traces from different public and private sources which include: (1) *network description files* with topological information, (2) *community databases* describing the interactions of users on the community web portal, (3) *period infrastructure and networking logs* from computational devices and (4) all messages exchanged between users on the 43 *mailing-lists*.

Each of the above sources required different analytical techniques for its analysis, based on the purpose of the experiment. In addition to the statistical modeling is worth mentioning fundamental methods of *structural analysis* (See 4.2.5) and *machine learning* (See 4.2.4).

4.1.2 Theoretical modeling

Game theory and mechanism design have been used to model the competitive/cooperation processes occurring on our scenario. Game theory provides a mathematical framework to describe these processes with functions in terms of rules, possible strategies and/or outcomes. These representations are not intended to be a faithful reflection of reality, rather than a model to capture the main characteristics under the study. Mechanism design, otherwise, intends to make the inverse process: from a desired outcome and a set of possible strategies, tries to find a set of rules that impose the outcome no matter which strategy the actors decide to play — by forcing them to play a specific strategy.

Section 4.2.1 provides the basic description of the game used in the simulations. Examples of mechanism design and evaluation can be viewed in Chapter 8.

4.1.3 Simulation evaluation

The regulation mechanisms developed in this thesis are intended for regulating scenarios with several properties typical from complex systems. The resource-sharing scenarios are composed by multiple free and non-trivial agents — nodes and resources —, sometimes interacting through multiple channels at once. Given the lack of access to a suitable common-pool resource to develop and implement our mechanisms, we used simulation tools to evaluate our proposals instead of a theoretical approach.

Naturally, although many of the results presented are supported by theoretical studies, their whole evaluation is limited to scenarios captured by our simulation model. The specific details of the computational and simulation models are described in Section 4.2.

4.1.4 Social network analysis

Social network analysis is considered a series of procedures and methodologies used to investigating social structures [15]. These structures are usually represented as graphs where vertices represent actors (e.g., companies, individuals) and edges represent relations between actors (e.g., friendship). The analytical methods developed have been used in mathematics and social science, but also in other research areas — like information systems [54] — where the structure under analysis can be represented as a network.

In this thesis we have used and contributed with new methods for social network analysis. Besides the generic methods for graph analysis we focused on detecting social structures (See Sections 4.2 and Chapter 7).

4.2 Methods and tools

4.2.1 Computational model

The theoretical analysis and simulations presented in this thesis are developed under several assumptions about the architecture of the nodes, their hardware resources and the software properties. This unified framework allows us to compare and combine the different solutions presented in the next chapters and to generalize most of the proposals.

Our scenarios are composed by a non empty set of N users $V = \{1, 2, 3 \ldots N\}$, each one contributing with one computing device i (e.g., a handheld, desktop computer, router) with a fixed amount of resources available to be used by itself and other nodes. The nodes can share their resources only with the neighboring nodes defined by the network topology $G = (V, E)$, where $E = VxV$ is a set of edges that connect pairs of nodes.

The type of resources to be shared are any type of hardware that can be shared in a real situation (e.g., CPU, memory, network interfaces). Therefore, the total amount of resources' slots available for a device i is represented by the non empty set $R_i = \{R_i^1, R_i^2, R_i^3 \dots R_i^M\}$. A slot is then, the minimum amount in which a normalized resource R_i^k can be divided and it will be our unit of work. While this definition is enough complete to represent any architecture for cooperative applications, its complexity also makes difficult the analysis of the results. Hence, unless otherwise noted, for the purpose of this work we assume that there is only one type of resource to be shared among nodes, usually noted as R_i. How to generalize our findings to scenarios with more than one resource is discussed on Chapter 8.

Applications, services or any other software that requires the use of resources is represented in our model indistinctly as a job. Each job has an associated minimum cost in terms of a set of resource' slots W without which the job cannot be done. The computational model assumes that nodes are able to add multiple heterogeneous resources from different sources without any penalty to fulfill the requirements of a job.

4.2.2 Simulation model

Most of the simulations in this thesis were performed with a non-cooperative game simulator [R12], a tool that allows us to configure a large number of complex scenarios based on the computational model described above. This simulator implements an iterative request-response model — similar to other n-players iterated games —, where nodes manage, ask and share resources according to our computational model.

While each experiment has its own configuration (e.g., network topology, number of rounds, nodes' strategies), the behavior of the nodes is always the same. At the beginning of each round, nodes are sorted randomly. Then, whenever a node needs to perform a job — decided with some random probability —, it first tries to use its own free resources r_i^k to complete it. If the node does not have enough free resources — meaning that $R_i^k < r_{min}^k$ —, then it asks its connected nodes for the remaining slots needed to accomplish its job, as describes the Algorithm 1. Fro now on, we will omit the superscript k as our simulations only use one type of resource.

Algorithm 1 Requesting procedure

Require: $G(V, E)$ ▷ Network graph
Require: R ▷ Set of resources
Require: i ▷ Requesting node
Require: j ▷ Answering node
Require: r_{min} ▷ Resource slots needed

$r_{ii} \leftarrow$ MIN$(r_{min}, R_i.free)$
if $r_{min} > r_{ii}$ **then**
    $X_i = \lfloor r_{min} - r_{ii} \rfloor$
    for all $j \mid (i, j) \in E$ **do**
        REQUESTRESOURCES(i, j, X_i)
    end for
end if

According the computational model described, the remaining slots needed X_i are asked to all neighbour nodes j simultaneously using the auxiliary function **requestResources**. As we discuss on Chapter 5 it makes the requesting nodes more agressive, but in return increases the probability to receive the minimum needed slots for performing the job. It is a common redundancy strategy in most computational distributed systems aimed to increase their robustness in front of node, network or task failures.

As we said, this procedure implements a non-cooperative model — similar to the Prisoners' Dilemma — which encourages the self preservation of resources over the cooperative actions. As the requirements for each potential job are in first hand supplied by the requester node i, it is probable that future sharing requests from other nodes will not succeed because nodes does not have much resources left.

After each node had the chance to start a new job, each of the nodes j who received a resources' request need to solve them using the answering procedure described in Algorithm 2. The procedure starts picking all the requests X_j received during the given round randomly and, for each of them, it decides to cooperate or not with a probability P (lines 2:3). This probability is the result of playing a contributory strategy (See 3.2.3) or one of the effort strategies developed in this work (See Chapter 5).

Algorithm 2 Answer procedure

Require: $G(V, E)$ ▷ Network graph
Require: R ▷ Set of resources
Require: i ▷ Requesting node
Require: j ▷ Answering node
Require: r_{min} ▷ Resource slots needed

1: $positiveReplies \leftarrow 0$

2: **for all** $X_j \in \text{RAND}(j.requests)$ **do**
3:     $P \leftarrow \text{PLAYSTRATEGY}(i, j, X_j)$

4:     **for all** $x \in X_j$ **do**
5:         **if** $P \leq U(0, 1)$ **and** $R_j < positiveReplies + 1$ **then**
6:             $positiveReplies \leftarrow positiveReplies + 1$
7:             $\text{ALLOCATERESOURCE}(i, j, x)$
8:         **end if**
9:     **end for**
10: **end for**

We want to emphasize that in our simulation model although resources are requested with the objective of accomplish a particular job, the decision of land or not those resources are taken slot by slot. It happens because despite that the probability P is calculated individually for each pair of nodes — taking into account the overall amount of slots X_j requested — the *Cooperate* or *Defeat* decision is taken for each slot x (lines 4:9), provided that the node has enough free resources (line 5). This allows us to design mechanisms with higher granularity, as we will discuss on Chapter 5.

The auxiliary function **allocateResource** simply allocates resources for a given job, without running it. Then, if a node has achieved enough slots to perform the job, these slots will be blocked for the duration of the job. However, any excess in the lent resources will be released, as describes the Algorithm 2.

Algorithm 3 Confirmation procedure

Require: $G(V, E)$ ▷ Network graph
Require: R ▷ Set of resources
Require: i ▷ Requesting node
Require: j ▷ Answering node
Require: r_{min} ▷ Resource slots needed

```
r_w = r_ii + ( Σ_{j=1}^{d(i)} r_ij )

if r_w < r_min then
    FREERESOURCES(r_w)
    return

else
    while r_w > r_min do
        if r_ii < r_min then
            j ← RAND(j | r_ij ∈ i.allocated)
            r_ij ← r_ij − 1
        else
            r_ii ← r_ii − 1
        end if
        r_w ← r_w − 1
    end while
end if

EXECUTETASK(r_w)
return
```

The confirmation procedure is executed on each node that decided to run a job the current round, to check if it has received enough resources r_{min} to perform the job (line 2). Any exceeding resources allocated are, then, freed using a random selection process (lines 6:14). Finally, if the allocated resources r match the requirements for the job W, the auxiliary function **executeTask** blocks the resources — reducing the amount of free resources ($R.free$) — on all nodes involved in the task for a period of time t, after which the main bucle will free them.

Notice that this model works under the assumption that users are not performing more than one task at a given time, which is not what one would expect in a real scenario. However, this design decision simplifies the analysis of the results and reduces the simulation time.

4.2.3 Stability of the models

The non-cooperative game simulator described above had been tested during the analysis of one of the early works [P5], using simple network configurations — topologies with just 2 or 3 nodes —, where participants had played deterministic strategies (e.g., answering always positive or negative to other nodes' requests). Additionally, the simulator code had been verified during its development using unit testing and internal assertions.

Each simulation-based evaluation presented in this thesis, with a particular set of initial parameters, is the result from executing the simulation for 250 rounds, but discarding the first 50 ones. Our early tests had shown that after 50 rounds, the simulations using contributory-based strategies were stable, but not in equilibrium: despite that the difference between averaged results for the evaluated variables (e.g., the ratio of cooperators) was less than 1%, nodes continue changing behaviours. Some simulations up to 10,000 rounds shown exactly the same behaviour.

In order to avoid random effects caused by the initial configuration of the simulated scenarios or the random variables used (e.g., the distribution of resources), each simulation had been repeated by default 100 times — and occasionally 1,000. Unless otherwise noted, when the results presented in this thesis as a distribution includes all the individual results in each of the simulations. In such cases that the results does not include a distribution, they are calculated after averaging each of the 100 individual simulations.

4.2.4 Feature selection algorithms

Features selection is an important step when datasets have many variables and correlated data. For an efficient identification of structures with data mining techniques, it is needed to evaluate and determine the variables in the data set that contain valuable information [55]. Determining subsets of a reduced number of features (e.g., with machine learning techniques) is another way to identify efficiently properties in data sets.

The feature subset selection in particular is a method for enhancing the performance of data mining algorithms by reducing the variables search space [56]. With this analysis we intend to: (1) identify a short list of features to understand resource-sharing processes in collaborative scenarios, and (2) evaluate each feature algorithmically using some well-known feature ranking algorithms.

Two well-known feature selection algorithms were used at some point: (1) *Correlation-based Features Subset Selection (CFS)*, an algorithm that evaluates the feature subsets, and (2) *ReliefF*, an algorithm that evaluates individual features. We have chosen these algorithms because they provide reliable feature sets, they are able to process continuous variables (e.g., ratios representing successful or unsuccessful process), and they let us understand the resource-sharing processes. Other algorithms can provide similar or even better results, but the underlying process is more complex to understand [55]. We briefly review these algorithms below.

- **Correlation-based Features Subset Selection (CFS) [57]**: CFS evaluates the usefulness of a subset of features by considering the individual predictive capability of each feature, along with their degree of redundancy. The algorithm selects subsets of features that are highly correlated with the class, but having low correlation between them.

- **ReliefF [58]**: This algorithm evaluates the usefulness of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different classes. We have chosen it because it is noise-tolerant and unaffected by feature interaction. However, ReliefF searches for all the relevant features, even if they are redundant. This algorithm assigns a “relevance” weight to each feature. In our analysis, we selected features with a relevance ranking above 0.

Instead of implementing these algorithms, for the analysis and evaluation of this thesis we used the Weka machine learning framework [59], which provided the implementation of these algorithms.

4.2.5 Community detection algorithms

Social relations in a group of participants can form a community if they are more willing to interact among them than with other members of the network. This is a well-known phenomena — called communities structure — that arises in most complex networks. The community’s size, structure or even members’ interactions outside and inside the communities are a good source of information to study the roles of the network users.

The community detection problem has been studied for a long time, and there are different algorithms and methods that can be applied, depending on the properties of the network and the properties of the targeted communities. In this thesis we apply two different methods, the *clique percolation method* and the *Louvain method* to detect two different community structures, and discuss the differences and the role of their members.

In practical terms, the difference is that while the percolation method is based on the detection and aggregation of k-clique disjoints sets inside the graph — which will have maximum connectivity among their members —, the Louvain method is an optimized algorithm to find partitions providing that the modularity (the relationship between average degrees inside the community and intra communities) is minimized.

- **Clique percolation method [60]**: The clique percolation method is based on the detection and aggregation of k-clique disjoints sets inside the graph, which will have maximum connectivity among their members. As the optimization is done locally, some members could be part of more than one community. Additionally, communities sub-graph are disjoint.
- **Louvain method [61]**: The Louvain method is an optimized algorithm to find partitions of large scale networks very fast, providing that the

modularity (the relationship between average degrees inside the community and intra communities) is minimized. As a result, we have a graph partition where all nodes are forced to belong to one and only one community.

4.2.6 Centrality measures

Centrality measures aim to identify the most important vertices in a network. The list of all possible centrality measures is too long to reproduce it below. However, next we provide the definitions of those centrality measures used in this work.

- **Closeness centrality**: Closeness centrality for a connected graph is defined as the inverse of the average distance to all other nodes (defined by their shortest path).
- **HITS [62]**: HITS is a ranking algorithm used in the past to exploit the web's hyperlink structure. As a result, it obtains two measures for each node, the *authority* and *hubs*. The first one is a measure of individuals as a source of information, while the second one ranks higher those nodes with high quality out-links.

4.2.7 Blockmodeling

Blockmodleing is an analysis method to reduce a large network to a smaller comprehensible structure. More specifically, according to Doreian [63], "blockmodeling tools were developed to partition network actors (units) into clusters, called positions, and, at the same time, to partition the set of ties into blocks that are defined by the positions".

Therefore, the objective of blockmodeling is to group or cluster the actors of a network by some *meaningful* definition of equivalence. As an example, *structural equivalence* [64] states that two nodes i and j are equivalent if they are both connected or not to the same other actors in the network, one by one. Then, the relations within actors in each cluster are used to build a simplified network graph, which is usually easier to analyze. Figure 3.2 shows an example extracted from [65].

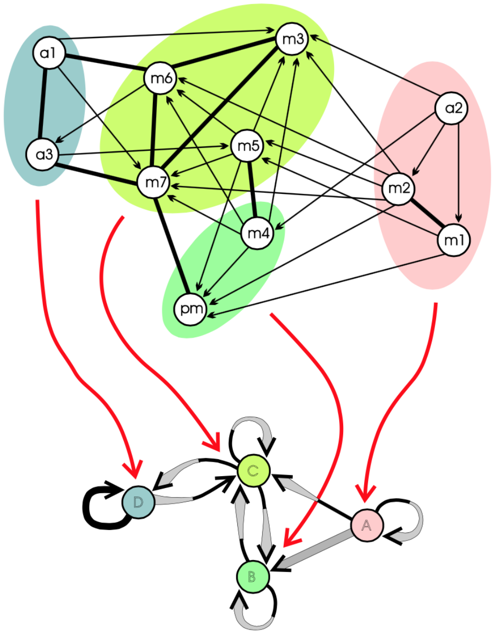


Figure 4.1: Blockmodeling example

Blockmodeling methods can be divided into two main categories according their criterion function: *indirect* and *direct* approaches. In the former ones, the measure of equivalence is a pre-condition, used to compute the similarity and dissimilarity of actors. Blocks are built then, using regular clustering methods. In direct approaches, on the other hand, the whole set of possible blocks — clusters — is tested against a set of perfect pattern of connections (equivalence). The uncertainty inherent in the indirect methods is compensated by the human interaction, as usually indirect approaches are supervised methods. It reduces the search space of solutions and the computation time.

Part II

Contributions

Chapter **5**

Effort-based incentives for resource sharing in cooperative applications

The effectiveness of volunteer computing paradigm depends on the collaboration attitude adopted by the participating users. Unfortunately for system designers it is not clear how to contribute with local resources to the shared environment without compromising resources that could then be required by the contributors. Therefore, many users or applications adopt a conservative position based on a direct reciprocity of actions (See Chapter 3).

This position limits the ability to compete for users with fewer physical resources despite their willingness and effort, while leads to an underutilization of the users' local resources reducing the efficiency of the ecosystem. In this chapter we address both problems by introducing a new reciprocal incentive mechanism based on participatory economic principles. The study discusses concrete implementations, which are analyzed in computing environments with both, abundance and scarcity of resources; and with perfect and imperfect information. The obtained results show how our mechanism is able to increase the overall balance of tasks executed without jeopardizing the availability of local resources.

5.1 Introduction

Volunteer and contributory computing are just two examples of the new emerging paradigms towards the future Internet, where communities of users aggregate their non-dedicated resources in a collaborative ecosystem. These paradigms have been used to support several types of applications, such as crowd-sensing [66, 67], mobile collaboration [68, 69] and grid computing [70, 71].

These computing paradigms typically involve a networked and distributed environment, through which heterogeneous devices share part of their resources (e.g., CPU or memory) to help other participants to perform certain jobs [1, 2, 72, 73] or running services [74, 75]. Although these applications are most often centrally managed, there are proposals that follow a distributed approach [76], with characteristics similar to Peer-to-Peer systems in which participants are both consumers and resource providers acting on their own interests.

Therefore, counting on a free and unrestricted access to the shared resources, without the need to contribute to the environment would probably lead to the collapse of the environment. We previously described this problem in Chapter 3 as the *tragedy of the commons*. Many previous proposals use reciprocal incentive mechanisms to promote collaboration among users, and thus dealing with the stated problem. These incentives regulate the global and individual benefits, by encouraging nodes to collaborate, granting them a *fair return* for their contributions.

Some research works argue that nodes contributing with more resources should receive in return a better service than those that contribute less [9, 42]. Hence, traditional methods typically use absolute metrics of contribution to determine what is a fair return and then, when a resource can be shared with other peers. While this approach is simple and efficient in most cases, it has been demonstrated that in heterogeneous scenarios typically discriminates nodes with few resources and does not provide a fair scenario for sharing resources [P2].

In this chapter we present a set of reciprocal incentive mechanisms for encouraging the exchange of resources in a highly heterogeneous scenario. The mechanisms are based on the proposal of [39] that uses participatory economic principles [7] for measuring the fair return of resources. In particular we are interested on implementing policies that *compensate the effort or sacrifice*, by measuring nodes' contributions as a ratio between the shared resources and the resources the node owns.

Therefore, the main **contribution** of this chapter is the **introduction of a novel reciprocal incentive mechanism based on effort measurements of nodes contribution**. The selected mechanism is the result of fine-tunning five different strategies based on the same concept of effort described early one. In order to determine the impact of using each strategy we performed several simulations that considered collaboration environments with abundance and scarcity of resources. The obtained results show that the effort-based incentives allow participants to accomplish at least the same number of jobs as when contribution-based incentives are used. Moreover, in the first case the nodes having few resources are not discriminated by the rest of participants, resulting thus in a more robust and fair collaboration process. Several features of the collaboration process are analyzed and discussed in this chapter in order to demonstrate that these findings are not by chance.

The next section presents the related work. Section 5.3 details the sharing strategies evaluated in this study. Section 5.4 shows and discuss the obtained results. Section 5.5 explores practical implementation issues and Section 5.6 presents the conclusions and the future work.

5.2 Related work

Incentive mechanisms for promoting cooperation have been studied on Peer-to-Peer (P2P) resource sharing and volunteer computing scenarios in order to convey selfishness behaviors from both: a theoretical and practical point of view. Theoretical approaches address the cooperation problem through game theory, typically using some variation of the prisoner's dilemma game [37]; e.g., the iterated prisoner's dilemma. In the case of repeated encounters among nodes (i.e., potential collaborators), positive interaction is very important for obtaining sustainable cooperation among participants [77].

The theoretical models have helped designers understand what properties must hold such incentive mechanisms to fairly allocate heterogeneous resources in a cooperative way. Two properties are especially desirable in the design of the node cooperation strategy; *incentive-compatibility* and *envy-freeness*, to impose a notion of fairness on the outcomes of every action made by the nodes. The first property means that players cannot improve their utility by lying about its resources. The second one indicates that players cannot prefer to use the resources of a particular participant. In [78] the authors established that at least one mechanism exists that holds both properties for agents with heterogeneous capacities, if there are only two kinds of shared goods or if the individual goods' values are binary. Another approach for providing fairness of effort-based strategies was proposed by Santos et. al. [79] who proposed the Quid Pro Quo (QPQ) mechanism to assign tasks on a self-organized and distributed scenario. QPQ also holds the envy-freeness property and the incentive-compatibility. However, as the previous theoretical models, it implicitly assumes that each participant scrupulously follows the specification of these mechanisms.

The practical approaches to address fair cooperation are mainly based on the proposal of BitTorrent [80]; a popular P2P protocol for file distribution. A key for the BitTorrent success lies in its Tit-For-Tat strategy, a reciprocity based mechanism, which works reasonably well to foster cooperation among downloading peers. In this scenario, some studies argue that nodes contributing more resources should receive in return better service than those that contribute less [9, 42]. These contribution-based mechanisms are simple and efficient in most cases; however they are unfair [81, 82], because some nodes end up contributing much more than they download.

In [9] the authors present a general heterogeneous model to evaluate the tradeoff between performance and fairness in BitTorrent systems. This work shows that the current protocol used in BitTorrent is only one of many possible solutions; in this case, the fairness metric is used to indicate incentive compatibility. The work also proposes three rate assignment strategies to optimize respectively one of the following variables: average downloading time, perfect fairness or max-min allocation. They also present a simple design knob that helps designers analyze the possible tradeoffs (in terms of performance) that the use of a certain rate assignment strategy has on the collaboration space. The performance is quantified in terms of both, average downloading time and fairness. A performance evaluation is finally conducted to show the merits and properties of BitTorrent-like protocols.

Similarly, the proposal in [83] focuses on the performance analysis of different policies using Tit-For-Tat. These authors introduce a model that characterizes the relationship between a peer's performance and the design parameters of the BitTorrent protocol. These parameters determine the effectiveness of the incentive mechanism for the nodes. This model is then used to determine how the incentive mechanism can be adjusted to enhance reciprocity or reduce inequity in the collaboration scenario. However, it has been demonstrated that when we apply contribution-based mechanisms in heterogeneous scenarios, the nodes tend to share and cooperate only between equals – in terms of resources [84]. Hence, nodes with scarce resources suffer discrimination because they do not have enough resources to contribute. While these works focus on a particular strategy, we analyze the performance of different collaboration incentives from a higher level.

Recently, a new practical scenario, named private BitTorrent communities, has appeared [85]. Such a proposal aims to motivate the resource uploading in P2P networks. Community administrators specify a minimum uploading threshold that must be addressed by the community members. In this way, it is guaranteed that each peer provides a certain level of contribution to the community. This mechanism, known as Sharing Ratio Enforcement (SRE), is very effective in increasing supply [86]. However, [87] show that SRE has two undesired negative effects: (1) peers are forced to seed for long times and (2) SRE discriminates against peers with low bandwidth capacity.

Although most of the BitTorrent incentives are contribution based, Rahman et al. [39] show that these systems can also be used with effort-based incentives. The use of such a strategy showed a greater utilization of the available resources and a reduction of the download times in slow peers.

In this work we apply the concept of effort-based in a different architecture, aimed for distributed computing instead of file-sharing. It means that the shared resources are not being transfered between users, only occupied — and blocked — for a fixed time. In bitTorrent applications, however, the files are transfered among users and the heterogeneous resources — bandwidth capabilities — are limited by both, the sender and the receiver.

5.3 System Model

5.3.1 Modeling Resources and Nodes Behavior

All the experiments performed in this chapter follow the computational and simulation model described early in Section 4.2. In this particular case, the resource requirements to perform a given job are randomly determined using a discrete normal distribution, originally with a mean of six slots and a variance of two slots — truncated to a minimum of one slot.

The number of rounds nodes need to perform every job is also randomly determined using a discrete uniform probability distribution, with a minimum value of one and a maximum of three rounds.

5.3.2 Resource Sharing Strategies

The request-response model implemented could be viewed as any iterative two-players game in the literature, like the Iterated Prisoner's Dilemma. In this game, two players are involved in a potential collaboration process choosing between cooperation and defection. The decision to choose one action or another can be based on the trust in others and their reciprocity. The consequences of the participants' choices have a different impact on the global community and also on their own probability to obtain resources from other nodes.

This study evaluates these impacts when the decision is based on two approaches: *contribution-based* and *effort-based* collaboration strategies. Next, we briefly describe the concrete implementation of the collaboration strategies used in these experiments, that we have used as a baseline.

- **Contribution-based Incentive**: This approach is implemented following the simple Tit-For-Tat strategy: a certain node i always responds to each slot request of a node j by making the same decision (i.e., cooperation or defection) made by j during the last request from i.

- **Contribution-based Incentive with Forgiveness**: To avoid the infinite cycle of mutual defection problem inherent in the TFT incentives, we decided to implement a second version where participants forgive a neighbor after a fixed number consecutive defeats. Particularly, when a node j has defeated node i during a consecutive number of rounds, the node i considers it as an unknown node.

We defined the **Effort-based Incentive** as a *reciprocity incentive mechanisms that measure the relative contribution* between nodes. In the simplest form, in this strategy the node contribution is defined as the percentage of available resources that a node shares with others. Hence, in our implementation a certain node i responds to the request of a node j, assigning to each requested slot a probability of cooperation P. The probability P is calculated as the ratio between the amount of slots that j shared with i in the last request, and the total amount of slots that j owns. That is:

$$P(W,t) = \frac{r_{ji}(t_{-1})}{R_j} \tag{5.1}$$

where W is the number of slots required for some job. Symbols t and t_{-1} stands for the current and last interaction between nodes i and j.

The metric is inspired by the work of Rahman et al. [39] in which authors rank and prioritize BitTorrent chunk requests from another peer, by using the percentage of bandwidth devoted to the other participant during a slicing windows of time. We adapted such proposal because in our scenario each slot is not transferable — in contrast of a BitTorrent schema where chunks (equivalent to our resource slots), could be obtained from multiple sources —. Hence, we have used jobs' slots instead of bandwidth to calculate the ratio between shared resources and the total amount of resources. In our strategy the ratio represents as a probability to cooperate instead of a ranking metric and the slicing window is just one interaction round.

The strategy implementation, however, assumes that nodes' effort have to be computed with high granularity — slots — instead of jobs. System designers could be tempted to simplify the metric to compute just jobs instead. Next strategies are variants of the first one to explore these effects.

- **Task Effort-based Incentive (one-by-one)**: This variation of the Effort-incentive measures contributions at job level, instead of CPU slot. It was inspired by the work of Santos et al. [79], where the authors discuss

— evaluating complete jobs instead of slots — how the granularity of their incentive affects the performance of the sharing strategy. In this case, the probability P is now calculated as the ratio between the mutual accepted jobs and the maximum of expected jobs that j could run. The mutual accepted jobs represents the sum of jobs that j accepted during last round from both node i and internal requests. The maximum of expected jobs is the number of j slots divided by the simulation average cost of jobs (in slots). That is:

$$P(W,t) = \frac{W_{ji}(t_{-1}) + W_{jj}(t_{-1})}{R_j \pmod T} \tag{5.2}$$

where T represents the average number of slots in the simulation.

Note that in this strategy we included the equivalent effort devoted to nodes themselves in the calculation of the mutual accepted jobs. It increases the diversity of probabilities P, avoids binary evaluations, and aims to maintain a reciprocity policy.

- **Task Effort-based Incentive (local)**: This variation instead of evaluating just the effort perceived by a particular node, evaluates the effort noticed by a sub-network cluster. Hence, the probability P is calculated as the ratio between the amount of jobs that j accepted from its neighbors and the maximum expected jobs.

$$P(W,t) = \frac{\sum_{j=1}^{neigh(i)} W_{ji}(t_{-1})}{R_j \pmod T} \tag{5.3}$$

As a consequence, the direct reciprocity between requester node and the requested cannot be assured, only the local one. The implemented strategy is a variation from the Task Effort-based (one-by-one) strategy, inspired by some ideas from network exchange theory [88]. This theory discusses, among other things, how the participants' outcome can be partly rooted in the structure of a (social) network, because they hold pivotal positions on the network, or in other case they are in a dependent position for responding to a request.

All unknown nodes, or those who have not met before (e.g., during the first round of the simulation), cooperates with a 50% probability.

5.3.3 Collaboration Scenarios

To study the impact of the proposed incentives, we consider two types of collaboration scenarios. In *scarcity scenarios*, the amount of resources available is — in average — twice the amount requested per node and round. Instead, in *resource abundance scenarios*, the available resources are four times the requested amount of resources.

Given the maximum amount of simultaneous jobs, the probability of having all resources occupied during a round is lower than 2% for an abundance scenario, and to 7% for a scarcity one.

5.3.4 Metrics

The simulation results were assessed using several metrics.

- **Node Cooperation Coefficient**: This metric is calculated as the ratio between the amount of requested slots by some node and the number of positive answers — slots — obtained during the simulation, regardless if the lent slots were used or not. It is, therefore a direct measure of the nodes' willingness to collaborate with each other.

- **Node Success Percentage (NSP)**: This value is the ratio between the number of jobs that a given node wants to perform during an experiment, and the number of completed ones. Therefore, it is a measure of the nodes' satisfaction.

- **Envy-fairness**: For a given node i with $d(i)$ neighbors, the envy-fairness metric is calculated as the difference between the success percentage of node i (NSP_i) and the average node success percentage of its $d(i)$ neighbors. Hence, it is a comparative measure of how envious a node is, considering its neighbors' jobs success. It is commonly referred as *node success correlation.*

  Notice that this metric pretends to measure a deviation of NSP between a node and its closed neighbors, rather than a property of the incentive mechanism — like *envy-free* —. However, we claim that while a positive *envy-fairness* of a single node does not imply *envy-freeness*, a negative one implies that the incentive mechanism will not be *envy-free*. In addition, all nodes using an incentive mechanism that is *envy-free* will have a non-negative *envy-freeness* by definition.

- **Tasks Fully Satisfied**: This value represents the ratio between the number of jobs that all nodes want to perform during an experiment, and the number of jobs effectively completed. Therefore, it is a measure of how many jobs get the amount of resources needed for their completion (which takes into consideration the temporal status of the nodes too).

- **Utilization of CPU slots**: We have here several sub-metrics related to the utilization of resources. The amount of *used slots* represents how many nodes assign to their own jobs. The *shared slots* indicates the slots devoted by a node to jobs of other nodes, and the number of *free slots* represents the slots that are not used. Finally, the *requested slots* show the amount of slots that nodes intend to obtain from other nodes.

The first three were measured directly on each participating node, and the last two considered the whole network.

5.4 Experimental results

In this section we provide a complete analysis of the experimental results to understand the impact of each incentive strategy (described in Section 6.4.2) in every collaborative scenario (i.e., in resource scarcity and abundance). According our models we considered only static network topologies — small-world or torus — with 125 nodes and heterogeneous resources distribution.

As an overview, Figure 5.1 shows the average cooperation coefficient — round by round — for contribution-based and effort-based strategies. We can see that all our mechanisms obtain more collaboration instances than the baseline ones. Otherwise noted simulations considered a resource scarcity scenario with a torus topology.

Focusing on the contributory-based incentives, we can see that, as we predicted, without forgiveness (i.e., replying strictly and reciprocally) all nodes start to

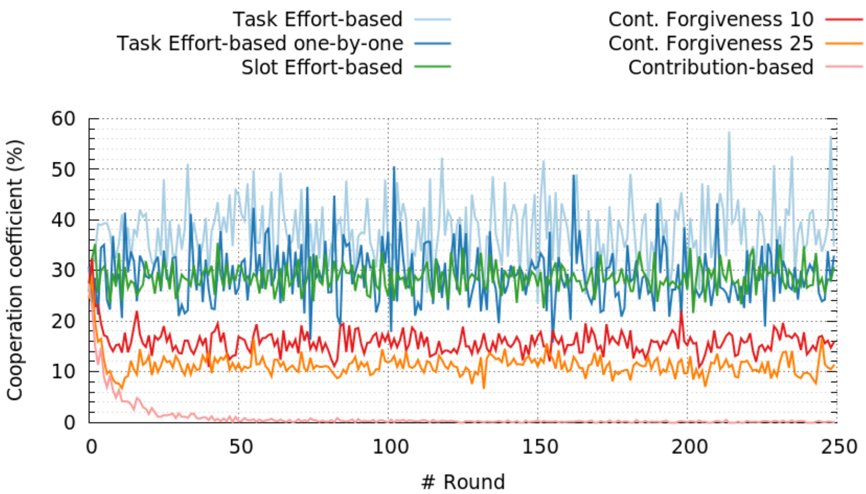


Figure 5.1: Average cooperation coefficient per round

reject requests of other nodes after the first 50 rounds. When a forgiveness variant is applied instead, the nodes are able to recover from this situation and cooperate between 17% and 11% of the time, depending on how many rounds they wait to forgive.

The remaining chapter will focus on determining if these initial findings can also be found in other collaboration scenarios and what are the consequences of using effort-based incentives instead of contributory-based on them.

5.4.1 Analysis of the Cooperation Coefficient

Figure 5.2 presents information about how heterogeneous is the cooperation coefficient of node in several scenarios — combining different topologies and resource availability. It shows high variations in the minimum and average values in all type of incentive, suggesting an explanation for the swings between periods with high and low cooperation shown before.

The results indicate that in all scenarios and topologies, the effort-based incentives obtain a higher cooperation level than those implementing contribution-based incentives. In our simulation, the reduction of 50% of available resources — from abundance to scarcity scenarios — represented a significant reduction

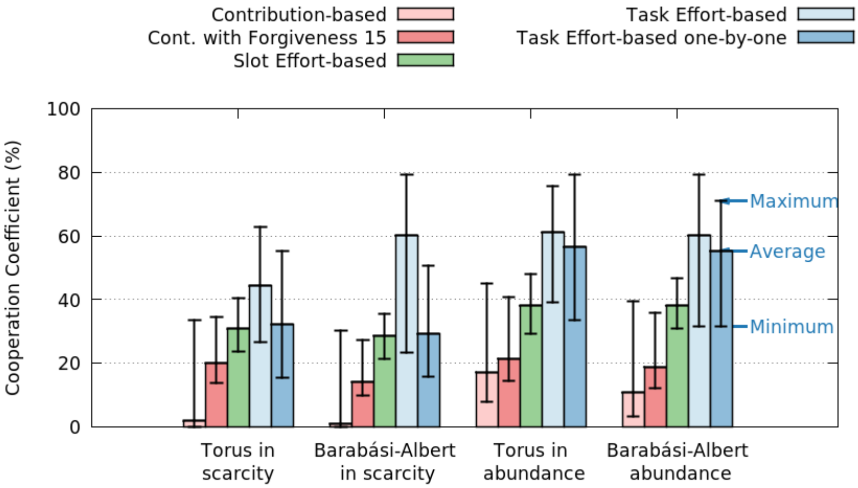


Figure 5.2: Cooperation Coefficient for torus and small-world networks

of positive responses; over 8%. According to that observation, **in scenarios where nodes use effort-based incentives are more interested in cooperation than scenarios where the contribution-based incentives are implemented**. However, it is also important to remember that the cooperation effort measures just a willingness, and not a tangible result.

5.4.2 Analysis of the Node Success

In cooperative applications, the effort is a measure, but not the objective. If an increment on cooperation does not imply a large number of jobs done, there would be no impact on users' satisfaction. Similar to that shown in Figure 5.2, Figure 5.3 shows the average, minimum and maximum values of the NSP.

These results support our initial hypothesis that **a participative measure of users contribution will lead to a higher cooperation, and therefore to a higher users' satisfaction**. This is a tendency observed by the average NSP, which is 6% over the satisfaction when nodes use a contribution-based strategy.

Node Success Percentage and Cooperation Coefficient show that **effort-based incentives measuring slots reaches the best scores in almost any**

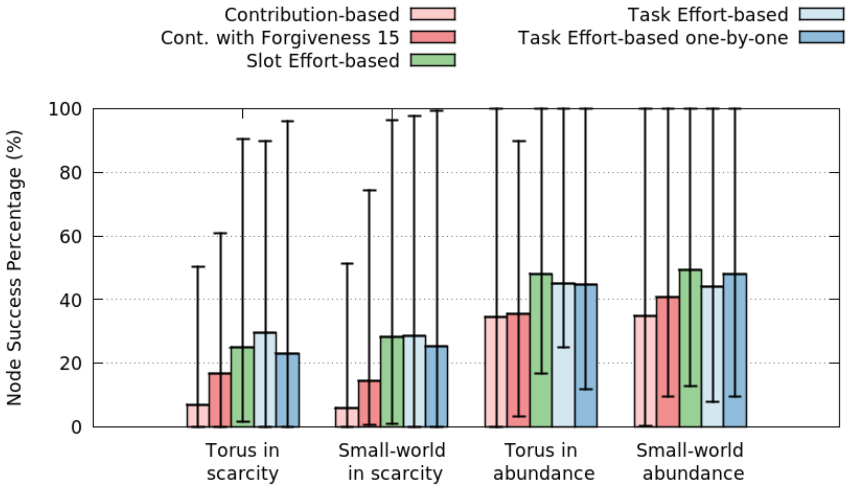


Figure 5.3: Node Success Percentage for torus and small-world networks

scenario. *The key element that allows us to understand this result is the granularity of the shared resources.* While in the slot effort-based incentive every slot is requested and shared individually, in task effort-based incentives the requested or shared resources are those needed to accomplish the job (i.e., blocks with a variable number of slots are requested and shared among nodes as a single unit). This imposes additional restrictions to the collaboration process. Therefore, **the collaboration process supported by effort-based strategies will be more effective and useful, when the resource being requested and shared is smaller**.

Figure 5.4 an Empirical Cumulative Distribution Function (CDF) plot for the NSP obtained by each participating nodes, with scarcity and abundance of resources, connected through a torus topology. We can see that nodes using effort-based strategies are involved in a fairer collaboration process; any percentile of population has higher NSP values, despite both kind of strategies accomplish the same number of jobs.

Evaluating the differences between both collaboration scenarios, Figure 5.4 shows that the total amount of available resources has a significant impact on the node success distribution. In scarcity scenarios approximately 60% of nodes have an NSP below 25% using effort-based strategies, while it drops to

5% when contribution-based strategy is used.

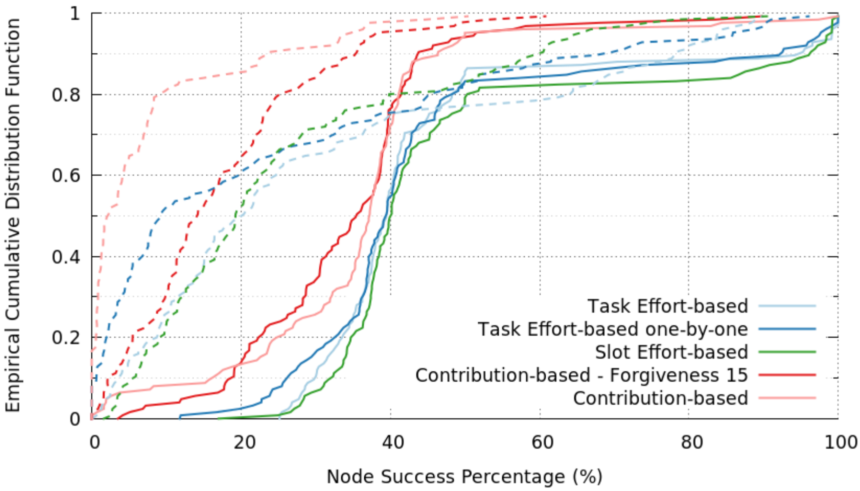


Figure 5.4: CDF of Node Success Percentage in scenarios with resource scarcity (dashed line) and abundance (continuous line)

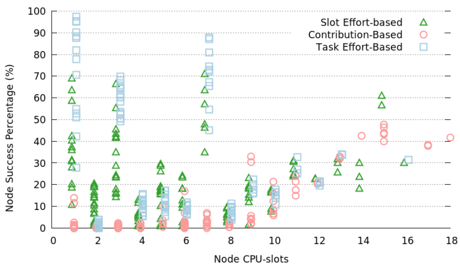
In abundance scenarios, however, there are almost no small NSP values. The percentage of jobs accomplished by each node, instead, are distributed very uneven with most of the nodes close to a 40% of jobs succeeded. This behavior of the cooperation process, particularly the stationary values, can be explained by the phenomenon of networks undergoing a phase transition. A Barabási study [89] describes this phenomenon, and shows that it is a common behavior pattern that appears when the network nodes are "socially related".

This phenomenon means that when a given threshold — called the tipping point — is reached, all the network nodes undergo a transition phase and they start acting as a single entity. In a previous study the authors shown that this effect appears when the nodes use a contribution-based strategy [P2]; however, Figure 5.4 also shows this phenomenon when the nodes use effort-based strategies. In terms of cooperation among nodes, this figure confirms the previous results.

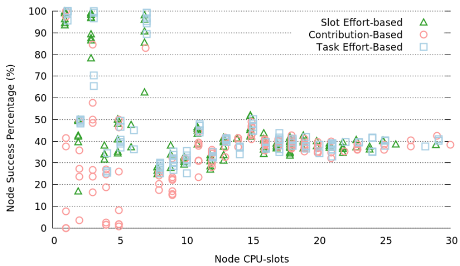
5.4.3 Analysis of the Impact on the Node Properties

Considering the previous results, we have to determine if the differences of performance between the collaboration strategies used in this study change with the node properties. The previous results show important differences between the maximum and minimum values of the NSP. This suggests that there exist node conditions or topological properties that help some nodes to achieve a better satisfaction than others.

The results shown until now discard the network as an explanation, since the torus topology shows almost the same behavior as the small-world[1]. Thereafter, we conducted an additional simulation to measure the individual satisfaction of the nodes. Figure 5.5 shows the NSP achieved by each node according to their total amount of CPU slots, in a scarcity and abundance scenario. The experiment involved a small-world topology and 125 nodes.



(a) Scarcity scenario



(b) Abundance scenario

Figure 5.5: NSP vs. Resources per node (R)

The results for scarcity scenarios show two different behaviors; one for nodes that have more than 8 resource slots, and another for nodes that have 8 or fewer slots. In the first case it does not matter what strategy they play, because both strategies achieve similar NSP values. However, when nodes have few slots, the contribution-based strategy is unfavorable. Contrarily, when using an effort-based strategy, the success percentage is between 3% and 70%. This shows that the **effort-based strategies are fairer than contribution-based strategies, when they are used in cooperative scenarios with**

[1]It is, however, a side effect of the topology's size, which is very small to have any effect. This and other topological related issues are the topic of Chapter 6.

heterogeneous participants (in terms of resources).

Although in resource abundance scenarios the results follow the same behavioral pattern as in scarcity scenarios, the nodes with more resources (i.e., large nodes) behave differently. In abundance scenarios the NSP of these nodes show a low dispersion, which means that they are almost self-sufficient; therefore, these nodes provide and request few slots to the environment jeopardizing the collaboration process and the task accomplishment in small nodes.

5.4.4 Analysis of the Envy-fairness

The Figure 5.6 shows the CDF plot according to the envy-fairness value recorded by the network nodes, and where all collaboration strategies show approximately 60% of nodes with negative — but small — values of envy-fairness caused by both, the strategies played by nodes and how we modeled the process.

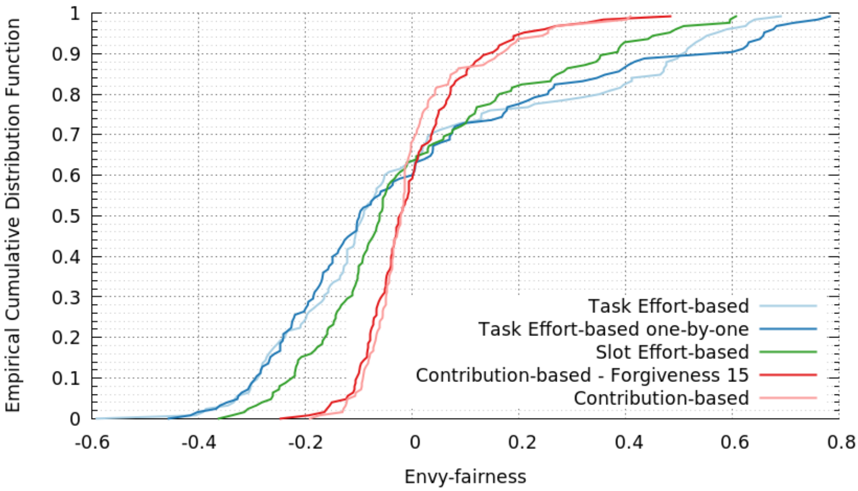


Figure 5.6: Envy-fairness in a resource scarcity scenario with torus topology

Analyzing the baseline strategies we can see small variations in terms of envy-fairness. This means that despite some nodes are not satisfied with their utility — when compared with their neighborhood — there is no higher inequality in the system. This result is not surprising, because the Tit-For-Tat strategy generates stronger cooperation dependence among nodes.

Effort-based strategies, instead, cause higher inequalities in favor of nodes with less resource slots as we can observe in Figure 5.7, where the envy-fairness value achieved by each node is shown. Nodes with less than 8 slots are more likely to have positive values in detriment of more powerful nodes; opposite as happens with contributory-based incentives.

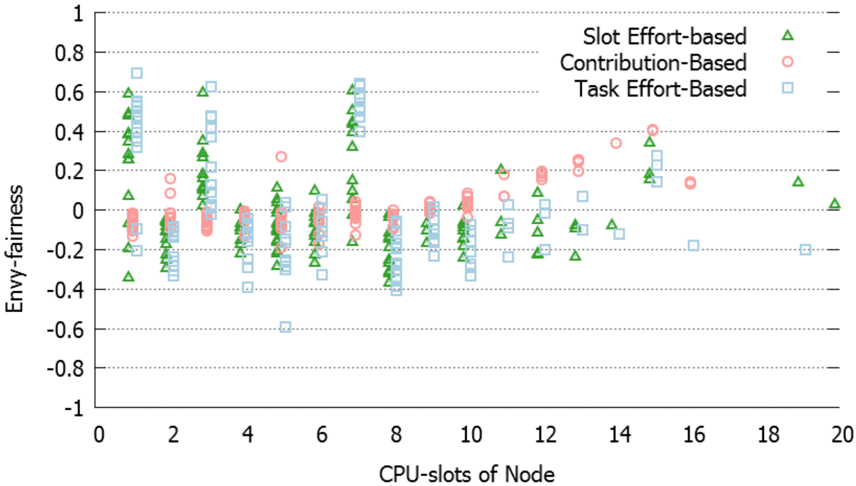


Figure 5.7: Envy-fairness vs. resource per node (R) in a resource scarcity scenario

The inequalities of resources distribution are an unwanted result from the strategy design point of view. Nodes with negative envy-fairness will surely complain about their status and would change their strategy if it would be possible. This effect is more noticeable in effort-based strategies — specially in those using jobs as a metric — due the higher values of some nodes' envy-fairness compared with the negative ones.

5.4.5 Analysis of the Resources Utilization

Another feature of the system that must be studied is the resource utilization. This metric provides an additional perspective of the fairness and effectiveness of the collaboration process. Figure 5.8 shows a histogram with the average distribution of resource slots in the six scenarios, using a torus topology. Four metrics are utilized to understand the resource utilization: *used*, *shared*, *free* and *requested* slots.

In resource scarcity scenarios, the results show that **nodes using an effort-based strategy dedicate at least the same slots to jobs belonging to other nodes as the slots assigned to local jobs**. The granularity of the measurement does not change this behavior. However, as we anticipated while measuring efforts at lower level — slots —, nodes tend to increase their solidarity towards other nodes, increasing the amount of resources used in proportion.

Furthermore, these results help us to understand the collaboration process embedded in scenarios with contribution-based incentives. It is our understanding that the eventual defection of some nodes causes an increment of free resources that nodes can devote to themselves, which as consequence causes more defections. At the end, each node into the system turns into self-provision, reducing the collaboration and efficiency (e.g., in scarcity scenarios there are about the double of free resources).

When node efforts are measured using slots, instead, nodes tend to increase their solidarity towards other nodes. This solidarity helps other nodes in getting all the required resources for their jobs, turning the competition on cooperation, making the resource sharing process easier, and thus increasing the NSP and the system utilization. These situations do not occur when a contribution-based strategy is used by the nodes.

In resource abundance scenarios, the cooperation improvement from effort to contributive strategies is smaller, and it also leads to a reduction of resources that nodes use for themselves. However, it **never turns the system to a point of solidarity, nor even neutrality**. This is an important observation about the scenario, as highlights the importance of studying *how* increase the available resources if needed; which we address in Chapter 6.

Finally, is noticeable that in all cases **there is a high number of slots underused by nodes**, which could contribute to increasing the value of the satisfied tasks metric. This side effect occurs because all strategies consider in their decisions only the previous relationship between the nodes. Any other factors, such as resource usage, energy consumption, continuous defection from

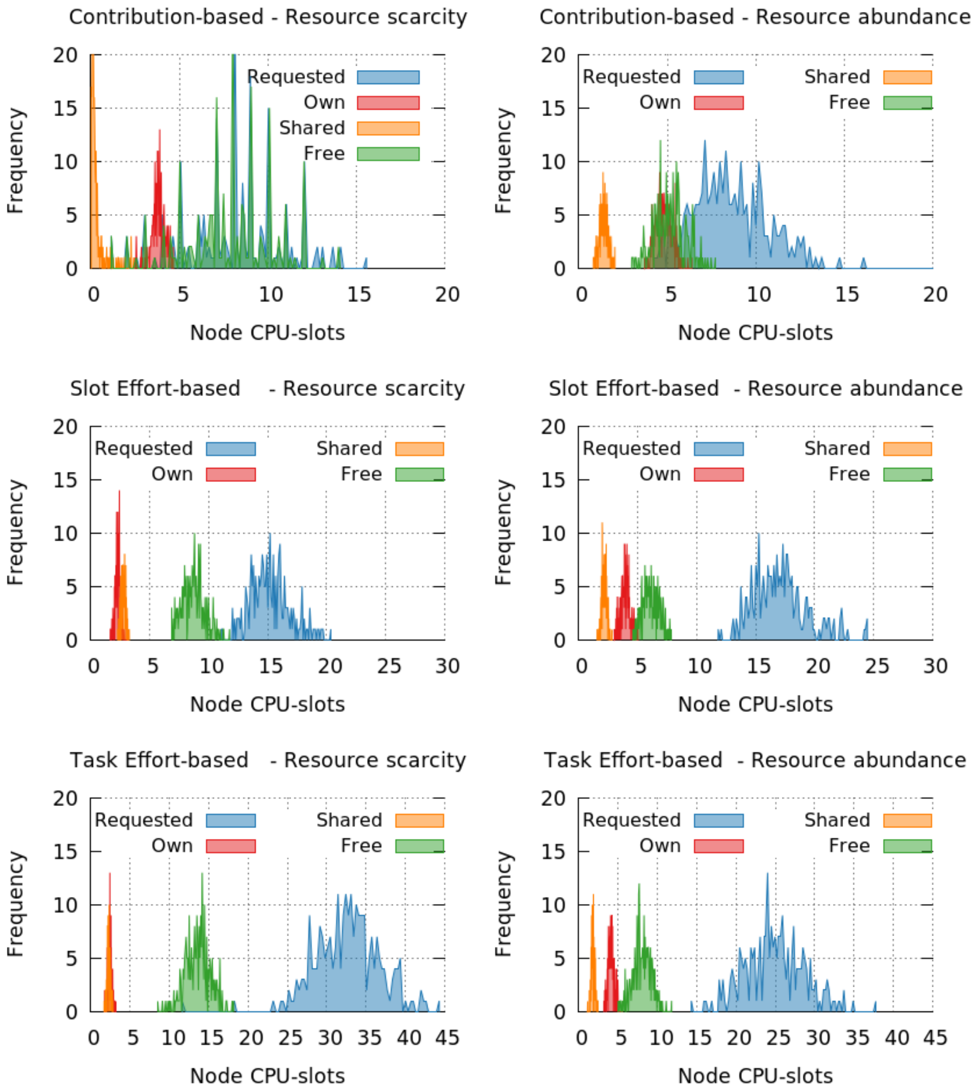


Figure 5.8: Histograms of average CPU slots requested, own, shared and free on different scenarios involving a torus topology

same neighbors or equitability, are not considered at this time.

5.5 Designing Effort-based Systems

The previous results allow us to understand the impact produced by the use of a particular collaboration strategy in a volunteer computing scenario. In this section we address some practical issues of the proposed incentives to guide mechanism designers.

5.5.1 Mixing Contribution-Based Strategies and Slot Effort-Based Incentives

One of the problems pointed early (See Section 5.4.4) was the undesired inequality of envy-fairness. We argued that a strategy enforcement through the mechanism design would avoid the strategy switching, as it happens in common-pool resources — where mechanisms and governing policies are enforced by the community. However, in most volunteer computing scenarios system designers cannot guarantee that all nodes will use some particular collaboration strategy.

Therefore, it is also important to understand the behavior of the cooperation coefficient when we have heterogeneity of collaboration strategies in the environment. Although that study is part of the future work, this section presents some preliminary results.

Figure 5.9 shows the cooperation coefficient and NSP when the collaboration environment involves different percentages of nodes playing contribution-based and slot effort-based strategies. For example, 40% / 60% means that 40% of nodes are playing the former one and 60% are playing the other one. These results indicate that the minimal and average value of NSP tends to improve with the percentage of nodes playing an effort-based strategy. Particularly, the collaboration environment shows two different behaviors depending on whether the percentage of the nodes playing effort-based strategies is below or over 60%. If the percentage of nodes playing the effort-based incentive is at least 60% — indicated in Figure 5.9 as 40% / 60% or higher —, the cooperation coefficient and the NSP tend to be similar. In addition, these values are always over the same metrics but for the other observed segment.

Although these results are still preliminary, **they indicate that if the cooperative computing environment allows the use of heterogeneous strategies, the system designer should try to ensure that the environ-**

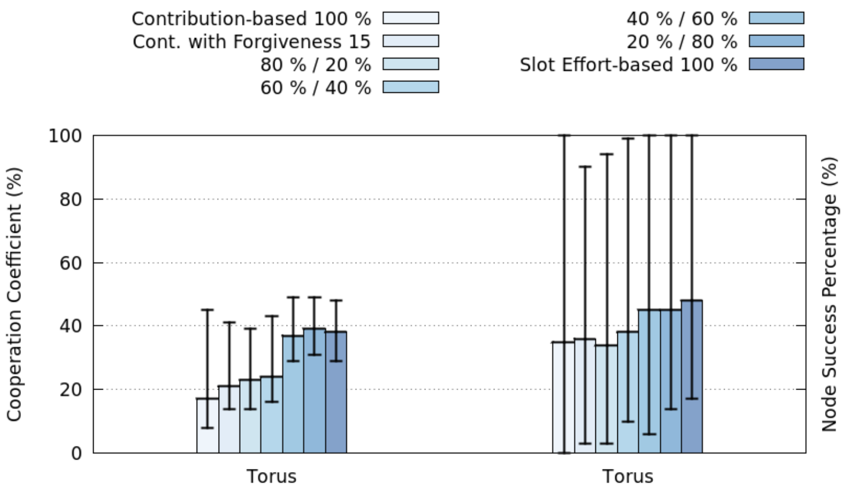


Figure 5.9: Cooperation coefficient and NSP mixing collaboration strategies

ment keeps at least 60% of nodes playing a slot effort-based strategy. That percentage seems to be the minimal threshold to ensure a good level of cooperation among nodes and also fairness of the cooperative computing environment. In a common-pool resources community, it also gives an idea about how much social pressure a misbehaving node could hold.

5.5.2 Analysis of Uncertainty on Effort-based Incentives

In practical terms, one limitation of effort-based strategies is that participants must somehow know the maximum number of slots — R_j — or jobs — $R_j \pmod{T}$ — that other participants can perform in order to evaluate their effort ratio. Two basic approaches can be used to address this issue:

- **Direct or indirect measuring**. This approach requires installing a specific program on peer computers to measure and report their maximum capabilities. However, this is costly in terms of processing time; therefore most systems use indirect measures to infer the capabilities of other nodes. The main problem with using inference is that such a process could be inaccurate [90], and therefore the system could promote the use of incentive mechanisms that are not suitable for such a scenario.

- **Node self-reporting**. In this technique, the nodes announce their maximum capabilities each time they make a new request for slots or jobs. Honesty is a key aspect here. Nodes with many resources could try to trick the incentive mechanism, announcing less resources than what they actually have, making the effort evaluation higher.

Cooperation coefficient and nodes' success percentage for effort-based strategies with a percentage of uncertainty — from 5% to 20% — in the measure are shown in Figure 5.10. By analyzing these results we can see small variations caused by the incorrect measure of nodes' maximum resources; making no difference in nodes' willingness to collaborate and the success of the collaboration process.

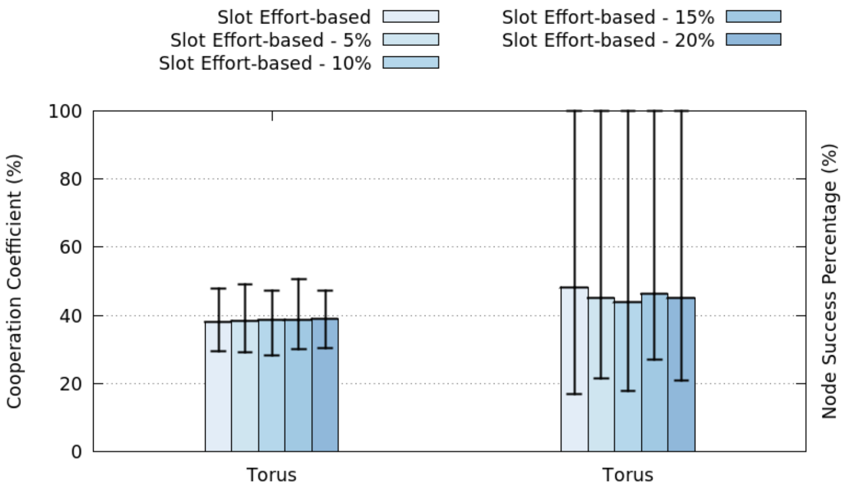


Figure 5.10: Results of study of uncertainty in the estimation of number of slots

The malicious behavior of some nodes, when they self-report their own limits, can be analyzed using the results shown as the NSP reported in Figure 5.11 and the envy-fairness of liar and trustworthy nodes in Table 5.1. During the experiment, we selected a percentage of nodes — from 5% to 20% —, named liars, which report less than their maximum CPU slots — from 1 to 3 in scarcity scenarios and from 1 to 6 in abundance — in order to be better evaluated by their partners.

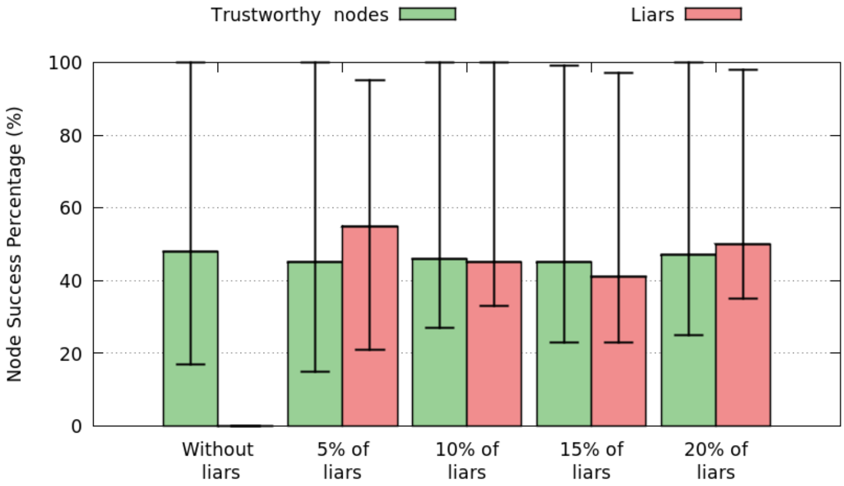


Figure 5.11: Node success percentage for trustworthy and liar nodes

Even if the liar nodes gain a better reputation during the first rounds, the results in Figure 5.11 show that there is no advantage for lying instead of truly reporting the node's maximum resources. Although there is a small advantage to liar nodes when they are a clear minority, it decreases — until disappearing — when the percentage of liar nodes increases. When the number of liar nodes increase significantly (e.g., above 20%) both type of nodes have the same envy-fairness.

Table 5.1: Envy-fairness of trustworthy and liar nodes

Strategy	**Co-player Cooperate**			**Co-player Deflection**		
	Min	Max	Avg	Min	Max	Avg
Slot Effort-based	-0.43	0.63	0.00	-	-	-
Slot-Effort based - 5%	-0.50	0.65	0.00	0.00	0.56	

5.5.3 Jobs management

Some results from Section 5.4 pointed out that most nodes are poorly evaluated by its peers due an increment of their requests, while the number affirmative answers they got has not changed. Additionally, we found that the experiments' design choices caused that all strategies — including contributive-based ones — have envy-fairness inequality.

This experiment has been designed to test what happen when the amount of slots requested during the experiment decreases. We can do that in two forms:

1. **Decreasing nodes' aggressiveness**: During the simulation, nodes are considered to be in two different states: active — meaning that the node has an own task running by himself — and waiting — when all own tasks has been finished. Each round, nodes that are in waiting status may decide to start a new own job with a default probability of 50%. We reduced this probability by half (25%).

2. **Increasing jobs' duration**: Each job performed in the simulation has a uniform distributed random duration from 1 to 3 rounds. When a node has an own job running, is on active state and hence, he is not requesting for new jobs to its neighbors. We increased such duration by a factor of 2 (from 1 to 6 rounds).

Interestingly, Figure 5.12 and Figure 5.13 show that increasing the duration of the jobs has no effect on the nodes' cooperation willingness. As a consequence, in both scenarios the average utility — NSP — remains equal.

On the other hand, a decrease of nodes' aggressiveness — by requesting tasks less frequently — has a negative effect, specially on scarcity scenarios. As nodes have more resources available, because there are less requests for sharing, when they want to perform a job is more likely to use their own resources than others. We have seen this effect before, when we discussed the behavior of Contributory-based strategy without forgiveness. As a final remark that would need further study, we can observe in Figure 5.13 that Contributory-based strategy without forgiveness is not influenced by jobs aggressively in abundance scenarios.

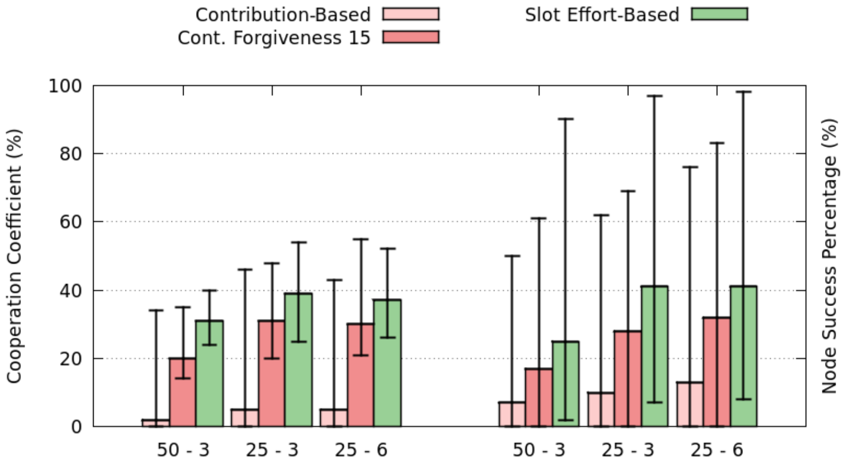


Figure 5.12: Cooperation Coefficient and NSP in a resource scarcity scenario

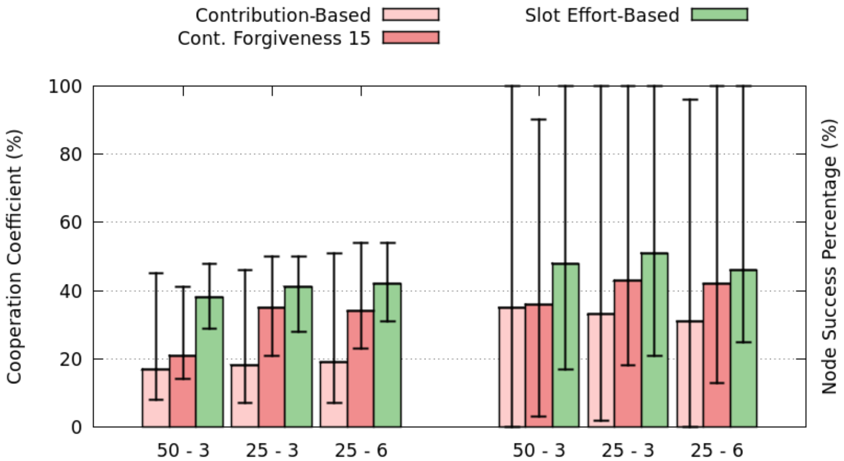


Figure 5.13: Cooperation Coefficient and NSP in as resource abundance scenario

5.6 Conclusions

In this chapter we investigate how to encourage the exchange of resources in highly heterogeneous volunteer and contributory computing scenarios by providing a set of new reciprocal incentive mechanisms. The mechanisms are based on participatory economics principles to compensate the effort or sacrifice, by measuring nodes contributions as a ratio between the shared resources and the total amount own.

The analysis and comparisons presented in this chapter provides evidence, based on simulations, indicating that it is possible to generously share resources with others without putting at risk the local resources using effort-based incentives. As a consequence, nodes showed a higher willingness to collaborate regardless of the amount of resources that they have, increasing the global ratio of tasks successfully finished. Furthermore, it avoids the tendency of cooperating only within nodes with the same amount of resources shown by nodes using a contribution-based strategy.

Looking at the results, we can conclude that the most benefited users from our proposal are the participants with scarce resources. It increases the envy-fairness of powerful nodes — thus, increasing the global envy-freeness — despite they are also benefited from the increase in cooperation willingness.

From the system designer point of view, we have evaluated the impact of envy-fairness when different strategies can be used, discussed the consequences of the uncertainty on the measurement or nodes' reporting of resources and the impact of several requesting policies. While the mechanism is far to be complete designed, the experimental results will be of great use for future works.

Summarizing, effort-based incentives are good strategies to be used in volunteer and contributive computing, since they help nodes with scarce resources to improve their satisfaction without harming powerful devices. However, it generates a feeling of envy among large contributors. Evaluating nodes' effort or sacrifice in terms of number of jobs will further improve the number of satisfied tasks per node at expenses of generating even more inequity.

Chapter 6

Compensating locality impact on resource-sharing scenarios

Despite the cooperation willingness shown by the participants on a volunteer or contributory computing scenario, the resulting outcome may be highly limited by the amount of resources they have to access. In cooperative applications, where resources are accessed only among pairs of nodes directly connected, the topological properties and nodes placement will influence the overall outcome of the shared environment. Hence, adding extra resources without taking into account their locality and accessibility will limit the desired impact.

In this chapter we first explore how much the global and local properties of networks affect the collaborative process after adding new resources into the system. The study outcomes indicate there is a short list of computing and network variables affecting positively or negatively the collaboration capability of the devices with less resources in such a heterogeneous scenario. Then, based on the lessons learned, we propose a heuristic to place extra resources in order to maximize the cooperation level in terms of resource sharing.

6.1 Introduction

Given their nature, the exchange of resources in distributed computing architectures — like volunteer and contributory computing — happens between users or devices directly connected to each other through an overlay network. As a result, collaborative applications must be designed to assure that any lack of resources can always be supplied by borrowing them from their neighbors. Therefore, mechanisms — like incentive strategies — are designed to guarantee the supply of resources locally rather than globally. Some of the results shown in Chapter 5 suggest that, as a consequence, the local optimizations cause an inefficient use of resources in the absence of other regulation mechanisms.

To solve the problem, most BitTorrent clients use opportunistic unchoking to compensate neighbors' noncooperation or lack of resources to dynamically modify their links [44], seeking to be connected to nodes with higher capabilities. However, in scenarios with insufficient global resources the continuous reallocation of nodes and jobs in the topology will not only not solve the problem, but also will lead to more inefficient systems.

One example of such scenarios are computer-supported mobile collaboration. This collaboration architecture involves nomad users with mobile devices and a software system to perform on-demand interactions [91]. For instance, the interactions among medical personnel at a hospital [92] or the incidents discussion conducted by construction inspectors after reviewing building facilities [68]. Typically small devices (e.g. cellphones) are well prepared to support tasks involving high mobility [93]; however they are the most critical resource in these scenarios.

Alternatively, *we propose combining traditional-resource computing paradigms* [17, 19, 18] (e.g., cloud or grid computing) *with the collaborative applications to create a decentralized collaboration network* [94], where *powerful* devices can share part of their hardware resources (e.g. processing power) during small periods of time in exchange for a retribution afterwards.

The resulting architecture would potentially have an increased heterogeneity, as cloud and grid computers have larger amount of resources compared with personal computers or cellphones devices. Hence, the location and distribution of these new resources will be crucial in order to improve both, the initial cooperation between devices, and the overall jobs successfully done. To that end, overlay topologies would play an important role.

After demonstrating the negative and positive effects of different network structures in the previous chapter, in this chapter we analyze how the locality of devices — their placement inside the network, compared with the other device — can improve or hinder the collaborative process. The main **contribution** of this chapter is the **description of a new heuristic to distribute resources and allocate their host nodes to cover the demands from powerless nodes**. The proposed heuristic will place devices with scarce resources in positions with easy access to the extra resources added, based on distance-based and local measurements.

Next section describes the problem to address and the research questions of this study. Section 6.3 presents the related work. Section 6.4 describes the experimentation setting. Section 6.5 introduces the hotspot placing algorithm proposed. Section 6.6 presents the obtained results and its discussion. Section 6.7 provides guidelines to deal with resource-sharing issues affecting the collaboration process. Finally, Section 6.8 presents the conclusions.

6.2 Motivation and problem definition

Finite resources and its heterogeneous distribution reduce the collaboration willingness of participants with less resources. In the previous chapter we have shown that this problem arises because small nodes — in terms of resources — cannot contribute with the same amount of resources to their neighbours.

As an example, consider a resource-sharing architecture with two types of nodes, *regular* and *powerful*; the former ones having about five times less resources than the later ones. Using the simplest version of a contributory-based incentive — that is, the Tif-for-Tat model described in Chapter 4 — we expect *powerful* nodes act as self-served, leaving the *regular* nodes few chances of cooperation.

For the sake of the demonstration, lets change the simulation model allowing nodes answer affirmatively as many slots requests as they want; without taking into account their resource limits. If then, we measure the nodes' average willingness of cooperate we expect the cooperation coefficient be raised close to the 100%; but not the nodes' success percentage, as the computation model prevents nodes for using more resources than they really have.

Figure 6.1 shows the results from this experiment, indicating that a hidden effect prevents nodes from achieving the expected cooperation. The simulation include 1000 nodes arranged according different topologies, being 60% of them *regular* devices and the rest *powerful* ones and average of 6 neighbors per node.

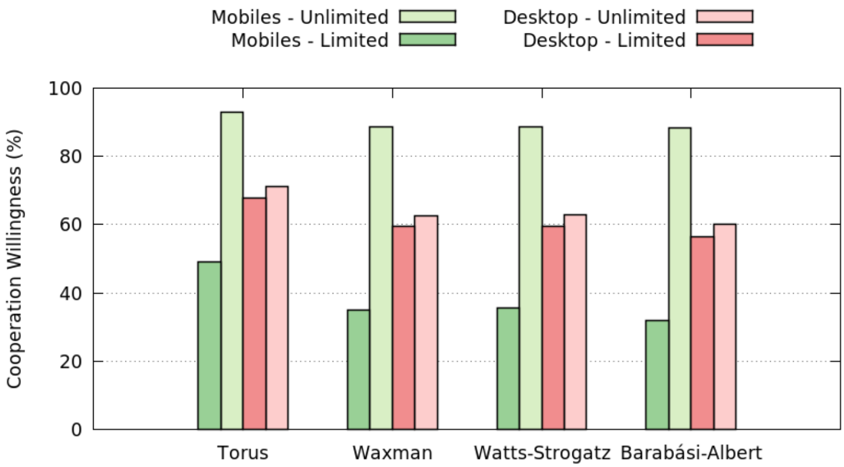


Figure 6.1: Cooperation coefficient of *regular* and *powerful* devices with limit/unlimited neighbor resources.

These results show that introducing *powerful* devices to the network and adding some limitations that did not originally exist in the Prisoner's Dilemma game, have a considerable impact on simulation results. In spite of having several *regular* devices willing to cooperate, many of them finally do not share their resources due to the limited number of available resources' slots.

More importantly, the variation of cooperation for *regular* and *powerful* devices between the limited and unlimited scenario is different for the four topologies, indicating that their connectivity influences the network process. These are statistically significant differences between the means determined by a one-way analysis of variance, ANOVA ($F(3, 996) = 2.614$, $P = 6.45e^{-25}$). Thus, these observations can be taken as valid.

In the results section we describe in detail why this collaboration process happens (See Section 6.6), and how we can take advantage of the underlaying reason to increase the number of success jobs performed by *regular* nodes (See Section 6.7).

6.3 Related work

Resource placement is typically a problem that tries to approximate the overall goal (performance or cost improvement), the workload and the target system. It is a NP-hard problem [95, 96], so it usually require some heuristics to find the approximate solutions within a feasible time. The first heuristic comes from the well-known uncapacited facility location problem [96], a widely studied quadratic assignment problem, where n facilities are assigned onto n sites, so that the average transportation cost between sites is minimized.

A well-known application domain of resource placement is the Web. There are numerous approaches for placing Web servers or Web proxies in a way that the performance is optimized [97, 98, 99]. The placement algorithms applied in all these cases use a global knowledge about the topology of the network and about the client requests.

Other approaches are presented in the domains of service grids [100, 101] and service overlay networks [102]. These systems also assume a global view of the network and some centralized management entity.

Finally, we also can find the same problem in a new area: service placement in intelligent environments (AmI). In [103] the authors propose a fully decentralized, dynamic, and adaptive algorithm for service placement in AmI environments. This algorithm achieves a coordinated global placement pattern that minimizes the communication costs without any central controller. Our work is different from this one in some aspects. A first difference is that in our solution every node can offer and request resources, whereas in the previous work the service follow a client-server architecture. Another difference is the use of the positioning element. In our case it is a generic resource and any node can offer it, whereas in the previous paper each client has to use a specific service.

Several approaches have been used to study the impact of overlay network topologies on the cooperation process; for example to analyze the topology properties in real-world applications with a good cooperation level. In that sense, Iamnitchi et al. have studied the patterns and properties of network topologies in file sharing applications [45], and Lozano et al. have studied them on email systems [11]. The topologies used in our study are based on these last two works.

A final approach was proposed by Nowak [8] who studied the properties of the topology for encouraging cooperation in the area of games theory. In his inspiring paper Nowak presents some mechanisms for the evolution of cooperation and simple rules specifying how natural selection can lead to cooperation in a Prisoner's Dilemma game. These rules have inspired our working hypothesis.

Studies by Cassar [10], Santos et al. [12] and Lozano et al. [11] also show the potential impact of the network topology on the cooperation process. Our study basically differs from these previous ones because we model the heterogeneity and limitations of computing devices. This study also takes into account the overlay network characteristics (e.g. the clustering coefficient and the degree distribution) and the placement of devices within the network.

Like these related works, we have limited the study to the ideal environment for the nodes' behavior. We then do not take into account other computational effects or drawbacks like load and task balancing or allocation. We do not take the nodes' mobility into account either, but we consider changes in the network topology. Simulating mobility of real-world nodes is a complex task [104], which should be addressed once the effect produced by the other network features has been understood. This study used data mining techniques to understand the process of collaboration among nodes and even to predict it.

6.4 System model

All experiments of this study were done with our network simulator following the same computational and simulation model described early on. Each experiment included simulations performed over a discrete scenario with 250 rounds, discarding the first 50 ones in order to avoid the transitory shown in Chapter 5.

6.4.1 Modeling Resources and Nodes Behavior

In these experiments, each computing device is modeled as being of one of two types: (1) *regular*: handhelds or similar computational nodes with few resources, and (2) *powerful*: laptops/desktop PCs or similar computational nodes with a larger amount of resources.

These devices were modeled as having only one type of resource to share and use (e.g., CPU cycles, memory, storage). We have set a ratio of 1:5 between the resources capabilities of this kind of devices. Consequently, we consider *small* devices — those initially forming the system — ones having up to 3 slots to use or share, while *powerful* ones — those introduced by system designers to supply the extra demand of jobs — will have up to 15 resource slots.

Finally, the maximum number of resources requested (W_{max}) was set to 10 and the maximum time execution needed for each job was established in 3 rounds.

6.4.2 Resource Sharing Strategies

These simulations were designed to asset the impact of topological-related decisions on resource-sharing scenarios, despite the particular collaborative process taking account. Therefore, we assumed that all participants have a simple sharing strategy — without need to further incentives to prevent free-riders or other malfunctioning behaviors —. As an example, in common-pool resource communities we can assume that nodes — users — complain with the incentive mechanism due some social pressure. The sharing strategy has been modeled using a contributory-based strategy — Tit-for-Tat — without forgiveness.

The simplicity of the model, however, made easier to generalize the experimental results and to integrate our placement algorithm on more complex scenarios as well as with the other effort-based mechanisms presented in this work (See Chapter 8).

6.4.3 Metrics

The simulation results were assessed using some of the metrics presented early on in Chapter 5 or a variation of them, like the *node reciprocity coefficient* or the *node failure percentage*. In their description, nodes and devices are used indistinctly as one node only represents a single device, and a device can only be attached to a single node.

- **Node Cooperation Coefficient**: This metric is calculated as the ratio between the amount of requested slots by some node and the number of positive answers — slots — obtained during the simulation, regardless if the lent slots were used or not. It is, therefore a direct measure of the nodes' willingness to collaborate with each other.
- **Node Reciprocity Coefficient**: This metric describes how much a node i is willing to cooperate with one of its neighbors j. It is based on the same principles as the Node Cooperation Coefficient — is calculated as the ratio between the amount of requested slots by i to j and the number of positive answers obtained from j during the simulation, regardless if the lent slots were used or not.
- **Node Success Percentage (NSP)**: This value is the ratio between the number of jobs that a given node wants to perform during an experiment, and the number of completed ones. Therefore, it is a measure of the nodes' satisfaction.
- **Node Failure Percentage (NFP)**: This value is the complementary of the NSP, and represents the ratio between tried and failed jobs during an experiment. The sum of both percentages, NSP and NFP, is one.

6.5 Hotspot Device Placement Algorithm

The placement algorithm has been designed for cooperative applications running on top of physical resource-sharing architectures, typically decentralized — without a central decision-entity — based on the observations described in Section 6.6:

1. It is important to maximize the number of links between heterogeneous nodes — hosting *powerful* and *regular* devices.
2. Devices with fewer capabilities — in terms of physical resources — have to be placed within the network topology in positions — vertices of the graph — with higher degree.

While the first observation imposes equal constrains for *regular* and *powerful* nodes, the second one describes a clear placement for *regular* ones.

In [98], the authors propose several heuristics to place resources in decentralized computing architectures. They characterize those heuristics by their *metric scope* — the nodes, resources and links that are considered when placing the devices — and the approximation method — the particular technique used to make the placement decisions.

The inherent decentralization and scalability that characterizes our scenarios, force us to consider heuristics that can be independently executed by every single participant. Therefore, we limited the metric scope of our solution to the information that one single participant has or can get very easily by just asking its neighbors. However, our ranking uses the knowledge about the overlay topology to compute the cost impact of placing one device on one specific location for all possible combinations (within the metric scope). These costs are then sorted, and the best one that does not violate any constraints is selected.

The ranking function of a location i in the topology can be defined as:

$$R_i(G) = dc(i) \sum_{j=1}^{neigh(i)} d(j) \tag{6.1}$$

where $dc(i)$ is the degree coefficient of the location or vertex i and $d(j)$ the degree of their neighbor j.

To guarantee that participants with fewer resources are not clustered together, if multiple vertices with higher ranking are directly connected our placement Algorithm 4 attempts to place first *powerful* devices close to critical positions, where their impact on the efficiency of the neighboring *regular* devices will be higher.

6.6 Experimental results

In this section we analyze the impact of increasing the resources heterogeneity on collaborative scenarios based on the overlay network topology by introducing

Algorithm 4 *Powerful* nodes' placement

Require: $G(V, E)$ ▷ Network graph
Require: S ▷ Number of powerful devices to place

$rankedPositions \leftarrow null$
for all $i \in V$ **do**
    $rankedPositions[i] \leftarrow$R$(i, G.neigh(i))$
end for

$counter \leftarrow 0$
for all $position \in$ SORT(rankedPositions) **do**
    if $counter < S$ **then**
        $G[position].type \leftarrow$R$(POWERFUL)$
    else
        $G[position].type \leftarrow$R$(SMALL)$
    end if
    $counter \leftarrow counter + 1$
end for
return G

extra resources. We firstly study the impact of adding *powerful* devices to a lower-resources scenario. Then, assuming that we found an optimal ratio, we further investigate the effects of topology in such scenario.

6.6.1 Analysis of nodes distribution

The bar plot (See Figure 6.2) describes quantitative the cooperation willingness of small participants in several resource-sharing scenarios after introducing new nodes with higher amount of resources. This result, has implications for distributing resources and selecting the most suitable network topology.

The resource distribution is represented in the *x-axis* (See Figure 6.2) as the ratio between *regular* and *powerful* nodes. For example, the first dataset was obtained with a network of 1000 nodes that had 20% of *regular* nodes and 80% of *powerful*.

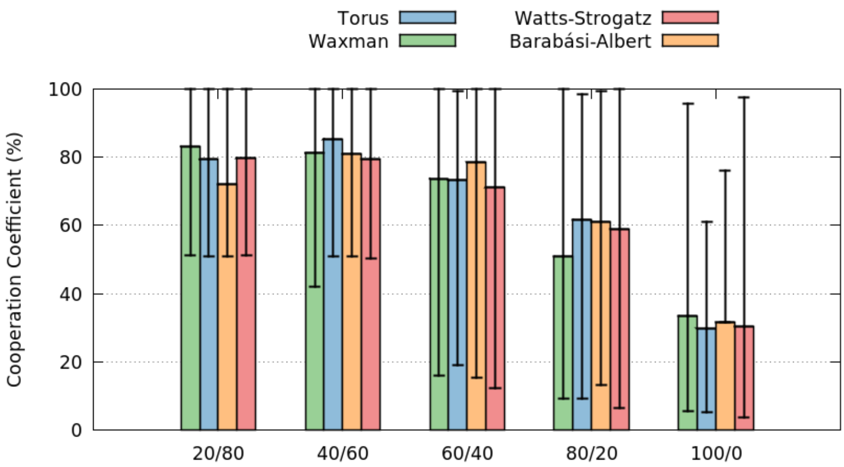


Figure 6.2: Cooperation coefficient of *regular* devices using four network topologies and several ratios of *powerful* nodes

We observe minor variations in the maximum cooperation coefficient due both, the addition of new resources and the overlay topological properties. However, Figure 6.2 shows how a small increment of resources in the scenario — from 0% to 20% — will increase the average cooperation coefficient in the *regular* nodes around 32%. These results confirm our hypothesis, showing that **the introduction of external resources improves the level of cooperation among *regular* nodes**. Most importantly, the scenario is benefited independently of the topological properties of the overlay network.

Increasing the ratio of powerful nodes from 20% until 60% will not affect the average cooperation, but it gradually increases the minimum cooperation until reaching the 43%. Adding more powerful resources beyond this point, however, does not grant any significant advantage to the *regular* nodes; but it will certainly increase the deployment costs.

These results show it is possible to improve the cooperation coefficient of the handhelds in a resource-sharing scenario with scarce resources by introducing new ones in the network. We used standard statistical methods to compute the confidence interval and margin of error [105] of the values. The standard error of the mean values is at most 0.055. The relative margin of error for the mean is at most 12.02% for a 95% confidence level. Thus, we can consider the average values computed on this test set as valid. The ANOVA ($F(4, 366) = 2.396$, $P = 2.49E^{-60}$) confirmed the significance of these findings.

6.6.2 Analysis of resources' distribution

Analyzing participants cooperation comparing different resources' distribution allows us to gain insight into the importance of locality. Consider, as an example, the introduction of a single node holding all the extra computational resources; instead of distributing them into several nodes. From the economic point of view this would be cheaper, but only few nodes would be benefited from the extra resources. Therefore, we are interested on finding, if any, a ratio of resources per node that guarantees the maximum cooperation profit.

Figure 6.3 shows the evolution of the cooperation coefficient in a Waxman network (see Section 3.3.1) when the amount of new resources is constant, but the distribution among the nodes changes from an initial ratio of 3% of new devices — with 112-128 extra resource slots each — to 33% — with 7-8 resource slots.

It is important to notice that there **exists a local optimal distribution of resources among the newly introduced nodes, that obtains a maximum cooperation coefficient** — from 20% to 27% depending of the metric

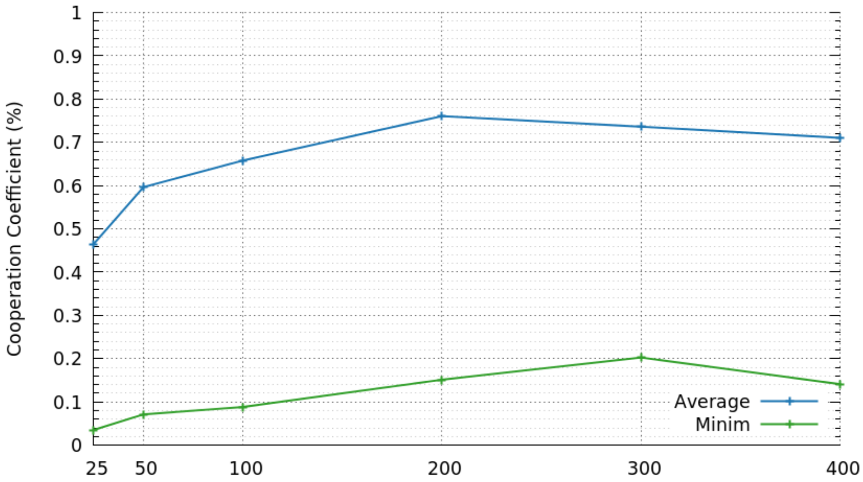


Figure 6.3: Cooperation Coefficient (Av, Min) of *regular* devices vs resources distribution ratio among *powerful* nodes

used (i.e. the average or the minimum values). The same experiment, repeated in other network models shown smaller cooperation differences, but maintains the maximum peak around the same ratios.

6.6.3 Cooperation reciprocity

Figure 6.4 shows the reciprocity coefficient between each pair of devices (x-*axis* represents the requesting node i and y-*axis* the responding node j) in a scale from 0 to 10000 where the stronger colors represents the higher values in a simulation using a Barabási–Albert network. Values below 0 represents nodes not connected among them by the topology.

As nodes have been ordered according to their node success percentage, we can easily observe that nodes with a higher cooperation —- on the right side of the matrix — do not correspond with the nodes that have higher reciprocity coefficient. In other words, **there is not a relation between how well a node is treated by their peers — the reciprocity — and the percentage of tasks he is able to finish at the end of the simulation**; which would be the expected result if participants are using a Tit-for-Tat strategy. **As a consequence, there should exist other better criteria (in terms of reciprocity and fairness) to place each device**.

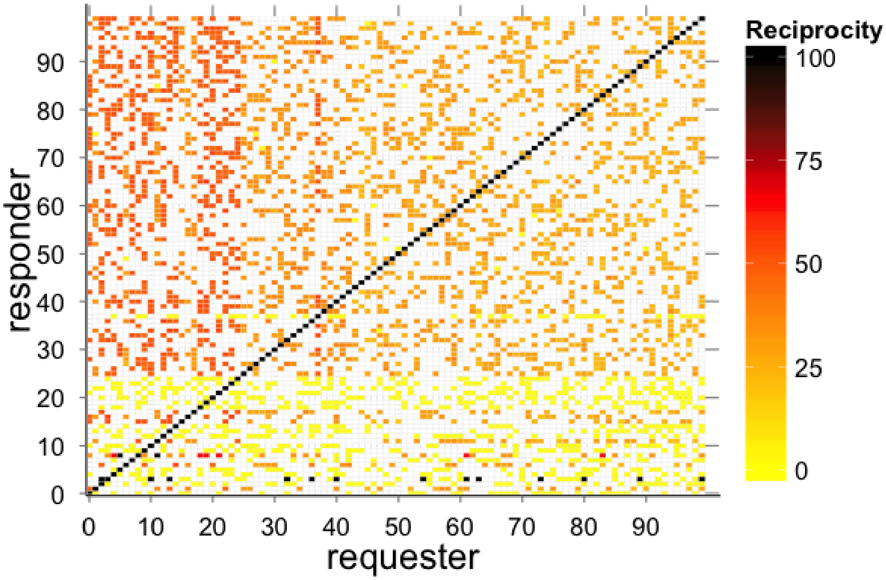


Figure 6.4: Reciprocity coefficient matrix ordered by cooperation coefficient

According to Figure 6.4, we can also observe that *powerful* devices are only focused on collaborating with a few number of neighbors between all the possible ones within the spectrum. To better understand this phenomena we need first to understand the role of the topology in the collaboration schema.

6.6.4 Impact of the network topology

In order to understand the effects of the network model, we are going to start reviewing the results presented earlier in this section regarding the impact of introducing *powerful* devices in a resource scarcity scenario; but from the topological point of view. The simulation included 1000 nodes — 60% of them *regular* and 40% *powerful* — arranged in different networks, all of them with an average degree of 6 neighbors.

Our measurement function is the Cumulative Distribution Function (CDF) of the *requests failure percentage* of *regular* devices — those with few resources —. This CDF value represents the percentage of participants that keep a certain defection level when trying to collaborate with their neighbors. The *x-axis* in Figure 6.5 represents the percentage of requests that failed because the potential collaborator had no available resource slots for sharing.

The simulation results using a Torus overlay topology show that all the participants have at least 40% of jobs failed because the lack of resources, while 60% of them only have a large NFP — around 78% or less —. Other networks, like Waxman, in which nodes are allowed to have different topological properties, shown a wider range of jobs' failures. It decreases the ratio of failed jobs for some portion of the nodes by depriving some participants to perform any task; like we observe in static random models like Erdős–Rényi and Waxman networks, where between the 29% and 18% of the nodes cannot perform any task.

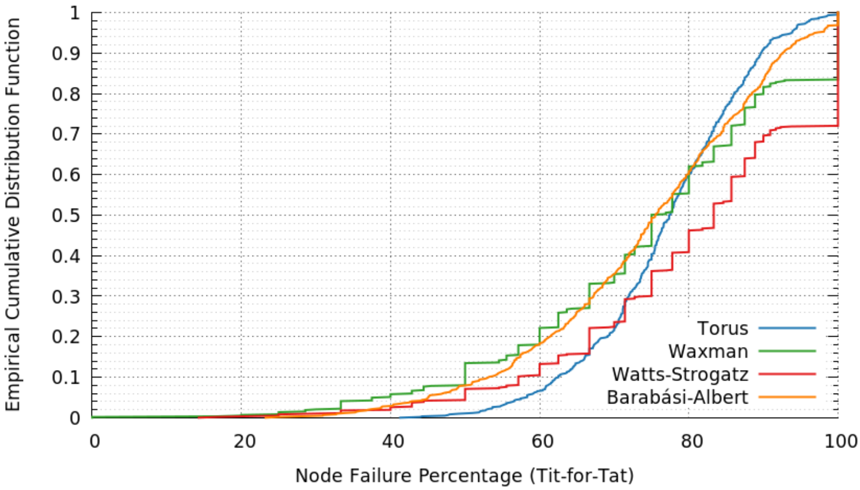


Figure 6.5: CDF of the failure percentage of *regular* nodes.

Participants distributed in topologies with a power law degree distribution and a smoother clustering coefficient (like Barabási–Albert models) have similar success and failure percentages as the ones participating in a Torus network. Furthermore, while its minimum failure percentage is not as good as the one observed on Waxman topologies, most of **their members below the 60th percentile have lower failure percentages than Torus networks, while the difference for the top 20th percentile is not as large**.

6.6.5 Impact of the network size

Most of the results presented in the previous and present chapters where analyzed in networks with an arbitrary number of nodes. However, we just show that the dynamics of the network may influence the cooperation between its members. One main cause can be the size of the network.

Our first hypothesis was that *if a node only interacts with its neighbors, modifying the network size without changing the average nodes degree — hence, altering only the clustering coefficient — should not change the nodes' cooperation response.*

However, the analysis of the cooperation coefficient for *regular* nodes in several random networks with different sizes (See Figure 6.6) demonstrated that there exists a propagation effect between nodes that modifies their behavior proportional with the network size. **While small networks assure a large percentage of satisfied jobs, nodes devices in large scenarios increase significantly their node success percentage; demonstrating that our initial hypothesis was false**.

If we now look at how resources are used by both types of devices (See Figure 6.7) in a small and large network with 125 and 1000 members respectively, we can observe that in the smaller network, both types of devices tries to get the fairest compensation for the shared resources because the Tit-for-Tat game strategy. As we have seen on Chapter 5, that is impossible for the *regular* nodes because they do not have enough resources compared with the *powerful* ones.

However, **when we increase the network size, devices with larger amount of resources are more willing to share their resources, dedicating less resources to their own tasks**. That benefits indirectly the devices with few resources, because now they have more chances to cooperate and share their resources. **As a consequence, *regular* nodes also devote less resources to their own jobs**.

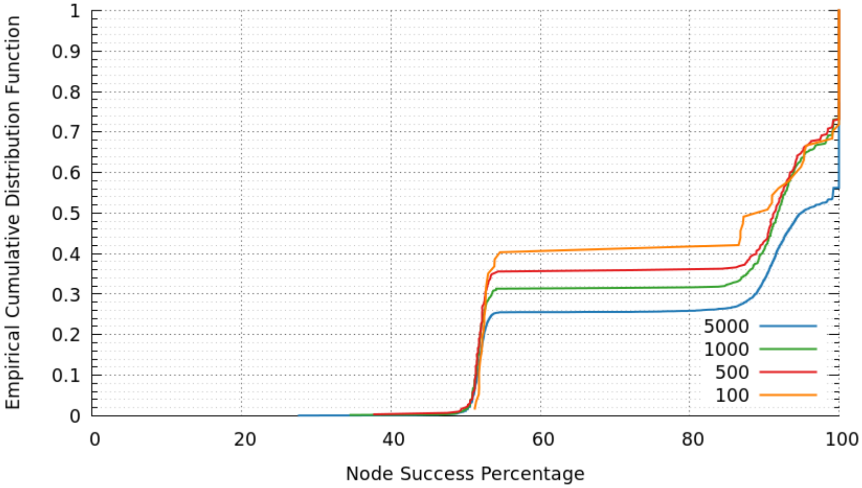


Figure 6.6: CDF vs. Node success percentage of handheld devices.

It's important to notice that in scenarios based on larger overlay networks, both types of devices tend to request less resources, as their probability of being not performing own jobs is higher than before.

6.6.6 Impact of the nodes' degree

If the resources needed for a given job cannot be obtained from the device itself, must be borrowed from their directly connected neighbors. Therefore, devices with higher degree would potentially have access to more resources, and will get a higher success percentage.

Figure 6.8 shows the node success percentage of several large networks — with 5000 nodes each — with 60% of *powerful* nodes. Two average values for the network degree were used: 15 and 35. The results show that for each tested topology **large clustering coefficients provide more collaboration opportunities and improve the overall satisfaction of the participants in terms of NSP**.

The results also highlights some other consequences of the collaborative dynamic implemented. Networks with larger nodes' average degree contribute to trigger the "networks undergo phase transition" pointed in the analysis of the scenario

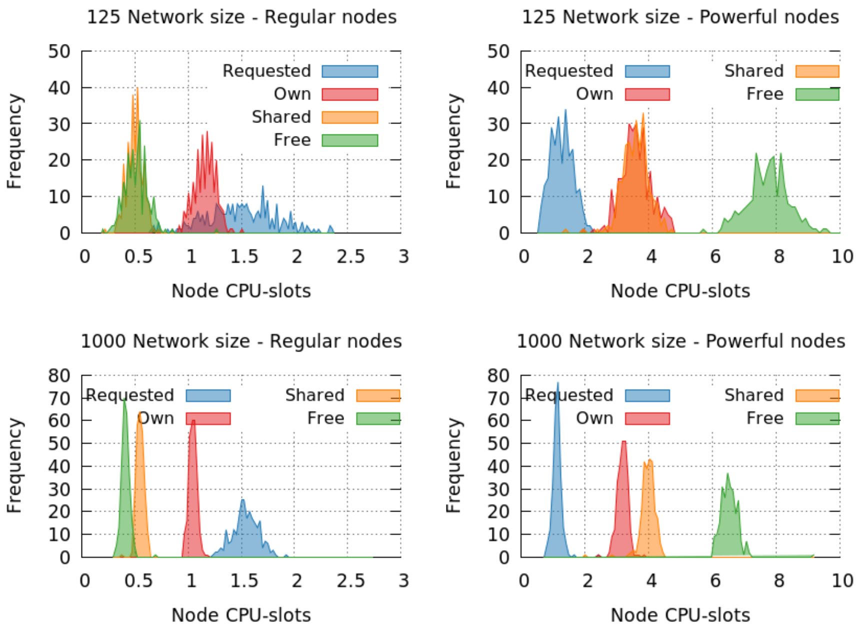


Figure 6.7: Histograms of average CPU slots requested, used, shared and free on Waxman topologies.

(See Section 5.4.2), creating more clear divisions between highly successful nodes and those participants with a 50% success rate (Specially in the NSP range from 50% to 83%). Although being something positive, it also contributes to increase the *envy-fairness* of the architecture.

6.6.7 Impact of other components

Understanding the collaborative process requires identifying if there exist any correlation between different features — the network properties analyzed and other components of the architecture. We have followed the steps and recommendations presented in [56] to choose the appropriate feature set and analyze it. Although there are techniques and algorithms to construct this feature set, we have created a set of “ad hoc” features because the data domain is already known to us.

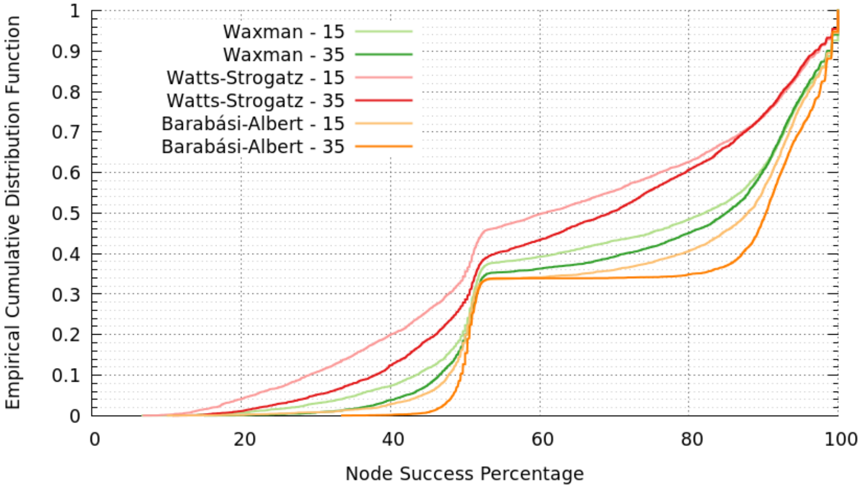


Figure 6.8: CDF vs. node success percentage of handheld devices.

- **Topological features** is the dataset that extends the previous topological properties studied. It includes (1.1) the *network* topology (torus, Waxman, Erdős–Rényi and Barabási–Albert models), (1.2) the *network size* in terms of number of nodes, (1.3) the local density of the nodes (*Clustering coefficient*) embedded in the network, (1.4) the nodes'*degree* of each node and (1.5) the sum of each *Neighbors' degree* at one hop.

- **Device features** is the dataset that represents the network participants and their behavior. It includes (2.1) the amount of available *resources* for each device, (2.2) the *device type* (*powerful* or *regular*), (2.3) the number of *rounds* in which a user has requested resources to other nodes and (2.4) the number of *requested resources* per job.

- **Proximity features** represent the relationship among nodes in the dataset. It includes some network distance metrics such as (3.1) the number of *powerful* nodes at one hop, (3.2) the total amount of *Neighbors' resources* per node at one hop and (3.3) the sum of *Two-hops neighbors' resources* from each node.

Note there are a couple of features that have higher correlation (i.e. similarity) between them. In that case it is enough for this analysis to use one of them.

For example we have not used the *device type* feature because it provides information similar to *resources*. In this case, we have chosen *resources* because it has a numeric value for the analysis. Likewise, we have used *rounds* instead of the *requested resources* feature.

We classified the datasets into four groups to show the relevance of the topology. Table 6.1 shows the results (i.e. the selected and ordered features) after applying the CFS and ReliefF algorithms to all datasets. We have used ten-fold cross validation in the evaluation of the features selection process [106]. For the CFS algorithm, we reported the number of times that a feature was selected in the ten-fold cross validation. For ReliefF algorithm, we reported the average relevance of the ten relevancies from the cross validation (selected features had a relevance ranking above 0).

Both algorithms selected the *rounds (2.3)* feature as relevant for almost every dataset. Notice the CFS algorithm does not select features with high correlation (e.g., it selects only the most relevant one between *powerful (3.1)* and *Neighbors' resources (3.2)*. Therefore the set of features selected by CFS tends to be minimal. However the ReliefF algorithm selects all relevant features, even if there is similarity among them.

A first analysis of these results indicates that the cooperation coefficient in torus and Waxman topologies depends mainly on proximity features due the lack of topological differences in the former one and the lower variation in the later. **In Erdős–Rényi and Barabási–Albert topologies, however, the nodes behavior depends on network topological features**. More specifically the cooperation coefficient depends on the clustering coefficient.

The set of relevant features obtained by both algorithms seems to be similar, but it cannot be clearly seen by just analyzing Table 6.1. In order to check such hypothesis, we re-processed the results obtained by the ReliefF algorithm applying Principal Component Analysis (PCA) transformations [106]. After applying the PCA transformations the obtained results indicate that the relevant features selected by both algorithms are similar, and they correspond to mainly those identified by CFS.

Table 6.1: Selected and ranked features by topology

Features	CFS			
	Torus	Waxman	Power law	Small-world
1.3 Clustering coef.			10/10	10/10
2.1 Resources				10/10
2.3 Rounds	10/10	10/10	10/10	10/10
3.1 Powerful		10/10	10/10	10/10
3.2 Neighbor CPUs	10/10			
3.3 2-hops neig. CPUs	10/10			

Features	ReliefF			
	Torus	Waxman	Erdős–Rényi	Barabási–Albert
1.2 Network size	0.001	0.001		0.002
1.3 Clustering coef.		0.002	0.001	0.004
1.4 Degree		0.009	0.007	0.002
2.1 Powerful	0.001	0.009	0.001	0.017
2.3 Rounds	0.001	0.001		
3.2 Neighbor CPUs	0.004	0.007	0.001	0.015
3.3 2-hops neig. CPUs	0.002			0.003

6.7 Improving nodes' placement

Previous results (See Chapter 5) have proved that devices with scarce resources are not as competitive in collaborative scenarios as devices with larger amount of resources. In Section 6.6 we analyzed some of the elements that influence — positively and negatively — such cooperation.

Some of the elements like the properties held by the overlay networks supporting the collaboration process, its size or the average degree cannot be modified in practice, because they are the result of the dynamic of the system under analysis or some external factors [R2]. Consider, as an example, the connectivity of several geographical towns using wireless technologies. In this scenario, the physical placement of the devices will be constrained to the geography; which will impose the network model and the optimal overlay.

Instead, the features selection analysis showed us that there are other properties — mainly related with the locality of the nodes in the topology — that could be easily controlled in order to increase the cooperation chances of *regular* devices. I our previous example, while we cannot control the underlaying topology, it would be possible to designate in which nodes we want to place more hardware — *powerful* devices — so that we compensate the lack of resources in *regular* nodes.

6.7.1 Using feature selection

The results obtained up to this point allow us to state that topological and proximity features are the most influential on the cooperation process. We tried to use this information to achieve an optimal network operation, by distributing the *regular* nodes according some topological properties but the results (See Table 6.2) shown no significant differences between the methods.

Since the cooperation coefficient is not sensitive to the analyzed parameters, it is our hypothesis that *besides the proximity features studied there exists another necessary structural property that triggers the collaboration of* regular *nodes.* In order to find it, we reviewed the experiments done so far using the most promising topological model; the Small-world networks.

A close observation of the results revealed that when *regular* nodes are placed following a degree procedure some of them end up surrounded only by devices of the same type. As result, these nodes have no access to the extra resources introduced in the network, and hence their cooperation drops. Comparing the cooperation coefficient on different placements, **we have detected higher reciprocity between heterogeneous devices than among homogeneous**. A logical reason for that phenomenon is that the introduction of *powerful* nodes mainly improves the cooperation of nodes directly sharing or getting resources from them. In our model these interactions are only allowed

Table 6.2: Cooperation coefficient of *regular* devices for various device placement strategies

Topology	Small nodes placed on	Cooperation Coefficient		
		Min	Max	Avg
Power-law	Clustering	39.89%	100.00%	79.93%
	Lower degree	40.76%	100.00%	74.25%
	Higher degree	37.79%	100.00%	80.91%
	Randomly	26.99%	100.00%	78.55%
Small-world	Higher clust.	49.40%	100.00%	81.67%
	Lower clust.	48.48%	100.00%	81.12%
	Higher degree	50.00%	100.00%	81.42%
	Randomly	47.78%	100.00%	80.77%
	Lower degree	39.11%	100.00%	80.40%
Random	Randomly	34.59%	100.00%	81.69%
	Clustering	39.89%	100.00%	79.93%

between participants directly connected to each other and, hence, the impact of *powerful* devices is mainly noticed when connected devices are of different type.

6.7.2 Using the Hotspot Algorithm

Using the feature selection for improving the nodes cooperation achieved by modifying limited nodes placement through the network ranking parameters (e.g. clustering coefficient and nodes degree) has been unsuccessful and not conclusive. However, the above experiments helped us to observe two conditions — necessary and sufficient — that, given our simulation model, and a hotspot placement problem; will improve the percentage of jobs done in the scenario.

1. It is important to maximize the number of links between heterogeneous

nodes — *regular* and *powerful* devices.

2. Nodes with fewer capabilities — in terms of physical resources — have to be placed within the network topology in positions with higher degree.

As we explained, the strategy of the algorithm consists on deploying first the *powerful* devices in positions of none interest for the limited ones; while keeping empty vertices for placing the devices with fewer resources.

Figure 6.9 compares the Cumulative Distribution Function (CDF) of the limited devices' success percentage on the same random network composed by 1000 nodes with an average degree of 30 and only 20% of *powerful* nodes. Each line corresponds to two different placement strategies: random and our hotspot algorithm (degree based).

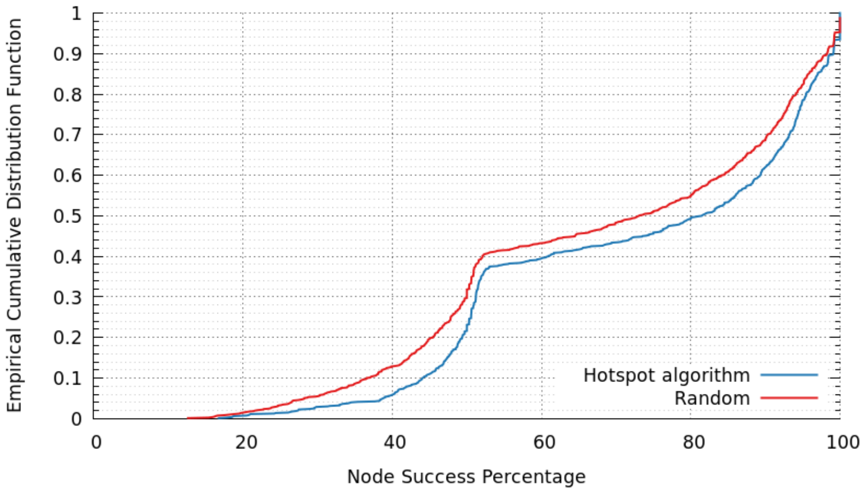


Figure 6.9: CDF of node success percentage random and hotspot placement strategies.

The nodes placement algorithm presented in Section 6.5 guarantees a sub-optimal distribution of *powerful* nodes, while increases the degree of nodes with less resources. As a result, it *increases the cooperation among heterogeneous devices and the overall node success percentage.*

Table 6.3: Cooperation coefficient of *regular* devices (Average, Percentage over 80% of cooperation) for random and hotspot placements in two network graphs of different sizes.

Topology	Nodes	Av. degree	Random placement		Our placement	
			Avg	> 80%	Avg	> 80%
Waxman	100	6	44.47%	14.63%	53.30%	20.00%
	100	30	73.73%	55.26%	73.03%	55.00%
	1000	30	68.65%	44.94%	72.73%	50.50%
Barabási–Albert	100	6	45.15%	8.43%	53.31%	22.50%
	100	30	76.54%	63.29%	76.00%	60.61%
	1000	30	75.24%	56.77%	77.69%	62.75%

The results for different network models and properties are illustrated in Table 6.9. The NSP improvement using our hotspot algorithm is similar while comparing sparse networks with similar properties (e.g., 8.83% in Waxman, 8.16% Small-world, with 100 nodes and 3 average degree). In networks with higher density (e.g., with only 100 nodes and an average degree of 30 nodes) both placement strategies generates similar number of links between homogeneous devices, and hence there is not a real improvement while our hotspot algorithm is used.

6.8 Conclusions

A number of issues related to the potential capabilities and limitations of devices with less resources in a collaborative application are understandable based on the simulation results shown in this chapter.

We have demonstrated that in a heterogeneous scenario — because the participants have originally different capabilities, or because we had introduced *powerful* devices to supply the demand — resources with scarce resources are found in a difficult position to cooperate.

Known data mining techniques were used to show that, among others, the aggressiveness of the *regular* devices — in the sense that they perform a high number of requests (See Chapter 5) — was one of the key factors. Considering the modeled scenarios, when the available resources are scarce, the collaborative applications should wait for a random time period before trying to submit new jobs. It addresses two situations: (1) to reduce the probability of collision of those requests and (2) to increase the utilization of shared resources. The new behavior will impact the Quality of Service (QoS) of users at short term, while will improve the network and computing efficiency.

Then, we have seen that the cooperation coefficient and the percentage of jobs finally done depends on the topology of the overlay network. In some scenarios we argued that it would be impossible to control how this overlay is built, while some times it may be forced by software design. In such cases, the software designers will be interested on implementing distributed algorithms to form networks with Small-world properties. Kleinberg identified the problem of how to find shortest paths in a decentralized way, in the case of Barabási–Albert network with only local information [107], which is the underlying idea for the work of Wang and Nakao [108], who propose a scheme of evolutionary game theory for topology evolution to change any given overlay topology into the Small-world structure.

Finally, the third component affecting the cooperation was the access to resources and its distribution among the different locations. On one hand, participants with fewer resources need in average more external resources than *powerful* ones. On the other hand, *powerful* devices are interested on cooperate in exchange for future favors. This is achieved by improving the connectivity of *regular* nodes and incrementing the amount of heterogeneous edges. We proved it experimentally, providing a hotspot placement heuristic that increments the percentage of successful jobs 8% under favored conditions (just 2% if the overlay topology has high density).

Chapter 7

Social effort on incentives

Many of the current most successful cooperative applications have been built around a community of practitioners and experts rather than by just private initiatives or stakeholders. However, the access to shared resources is still regulated by traditional incentive mechanisms, which are usually based on participants' contributions or resource sharing ignoring other non-technical activities that members voluntarily perform to keep the community alive (e.g., administration and coordination of the tasks). Therefore, some community members' who have played an active role in supporting the development, maintenance and growing of the ecosystem may feel unfairly treated and discouraged to continue contributing if their effort is not recognized somehow by the community.

In this chapter we focus our attention on measuring the effort of community members in non-technically related tasks by analyzing their interactions in common social venues, like mailing lists. In practice, it required first a) to test some structural analysis methods and b) propose a new framework to detect positions and roles in more complex social structures, like multi-relational and multi-layer graphs. This chapter is our first step towards the design of incentive mechanisms rewarding non-technical contributions on cooperative applications.

7.1 Introduction

Like most large scale and distributed systems aimed to share resources in a competitive environment, cooperative applications must be built upon a supporting architecture with governing mechanisms that guarantees they can operate efficiently. In the previous chapters we have discussed how to build mechanisms more inclusive by a) measuring nodes' efforts instead of their direct contributions when they are sharing computational resources (Chapter 5) or b) by introducing new computational resources in specific locations to increase the cooperating opportunities of nodes with fewer resources (Chapter 6).

Both previous mechanisms are intended for scenarios where users can share their superfluous computational resources, but they can also be easily adapted to other common-pool resources. One characteristic of such scenarios is that users are organized together as a community to guarantee that the ecosystem works properly, instead of depending on some single authority.

Although the main purpose of the community is to share some good, the tasks of community members also include other supporting activities necessary for the growth and improvement of the ecosystem (e.g., coordinating meetings, accounting). As an example, in volunteer and contributory applications communities the main activity of the participants is to share computational resources in exchange for others, while the supporting activities might include developing or improving the software, helping new members or simply taking care of the website.

Traditionally, the governing policies only paid attention to the time and effort devoted to the main task of the community members, without considering that the supporting activities are also very important. This difference of criteria can cause a detachment from users towards the collective, creating a lack of human resources which will lead to an inoperative common-pool resource.

However, under participatory economic principles, users — and more specifically, their time and effort devoted to the community — are also important measurable and accountable resources. According to this idea, our framework will measure the supporting activities performed by each user and report them as a *social effort*, which will grant similar rewards as sharing resources with others (a retribution in form of resources by their peers). While the mechanism had created for compensating users already performing supporting tasks, it is expected that other users will join them as now they have an incentive to do so.

In this chapter we narrowed the social venues under analysis to just on-line participatory forums, which allows us to easily track users' interactions. The utility of users' participation has been measured, hence, as its influence across the graph of relations with their peers on the different participatory forums.

Therefore, the main **contribution** of this chapter is the **introduction of a new framework to find positions and roles using comparisons between actors and sets of actors instead of just using pairwise comparisons**. In this way we enable the usage of many more measures of similarity inside position and role detection methods (e.g., based on distances, community structure, triangles and cliques). As a result, we can identify new types of easily interpretable positions in complex graph structures like hypergraphs or multiplex/multi-relational networks (which is needed to represent the interactions among users in their common social venues).

We have evaluated our work on both synthetic and real data, using several existing and new similarity measures and providing both qualitative and quantitative evidence of the new possibilities enabled by our approach.

The next section presents a preliminary study motivating the development of a new framework for identifying role and positions. Section 7.3 describes the state of the art on blockmodeling systems for multi-dimensional data, while Section 7.4 revisits the main concepts related with blockmodeling. Section 7.5 describes our extended framework. Section 7.6 shows and discusses the obtained results. Section 7.7 presents the conclusions and the future work.

7.2 Motivation and problem definition

Creating a *social effort* measurement that reflects the users' involvement and participation in community supporting activities is a challenging task. Almost any activity performed by some user that is related with the community could be — potentially — considered as a contribution. Nevertheless, not all the possible actions are an effort worth of being considered a useful contribution for the community or to be rewarded.

Given the impossibility of defining and measuring something as subjective as what useful contributions are, in this work *we focus our attention on on-line social venues and participatory forums because their nature make easier to collect information, while provide a good representation of the relative influence of users.*

7.2.1 Measuring users participation in supporting activities

In order to have a brief idea about the challenges we need to address while measuring the participation of users in such a complex system, let's study the differences between users' participation in the *main activity* and *supporting activities* in another context similar to ours. Figure 7.1 shows the empirical cumulative distribution function (ECDF) plotted as the Lorenz curve, of users participation in guifi.net [26], the largest community network [P1] to the best of our knowledge. The participation is measured separately as the number of new devices created by users — one of the main technical activities of network communities — and the number of messages exchanged in one of its participatory mailing list — which is a reflection of social interactions among users with the aim to share knowledge or help other members.

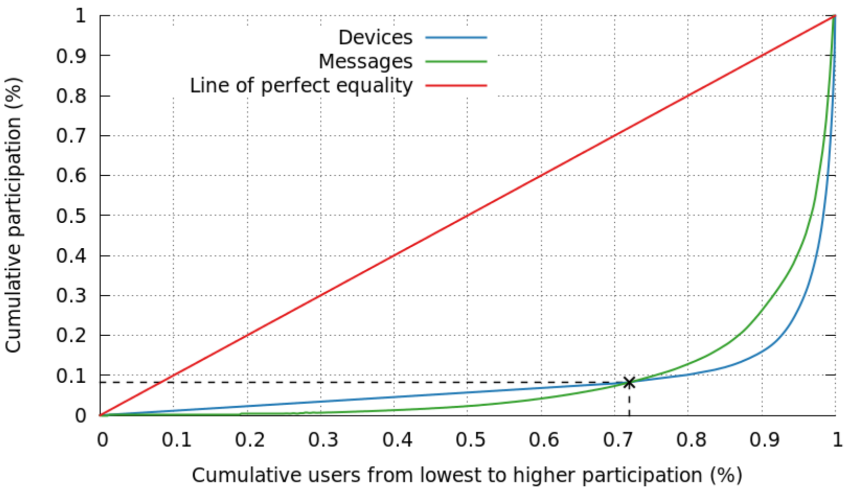


Figure 7.1: Gini coefficient of two participatory forums in guifi.net.

The Gini coefficient [109], measured as the area between the line of equality and each of the curves, is close to the absolute inequality in both activities — 0.8358 in the devices creation and 0.8320 in the message exchange. Most importantly, the Lorenz distribution function also suggests that network members behave differently in terms of participation in the examined forums.

It is not unusual that there exists more than just one *social venue* or *participatory forum* helping users to develop their supporting activities in the same community. In the context of community networks, users might have a mailing list to discuss general topics concerning the community and other ones more generics to discuss software development efforts. As individuals, the community members can show also different contribution patterns on each of the participatory forums.

Figure 7.2 shows this effect by plotting the number of messages and devices created by each user identified in two key mailing lists: *users-list* and *dev-list*.

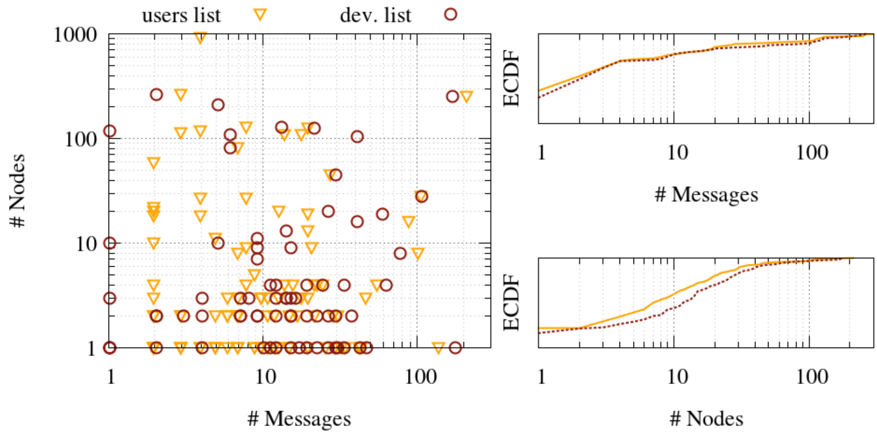


Figure 7.2: Guifi.net users participation

We observe that most of the users are selective and choose to collaborate only on one kind of activity (main or supportive), contributing with little or nothing to the other. For example, there is a high concentration of users participating in the development mailing list, but these are users which contributed only with one or two devices to the physical communication network. Except for a few members — founders or evangelists — **users tend to choose to collaborate only in one task of the community either the main one or supporting ones**.

Notice also that users have heterogeneous levels of involvement on each participatory forum too, making it harder to measure their effort in the overall community.

7.2.2 Ranking users' participation in supporting activities

The second important challenge we need to address is how cluster users with similar levels of contribution, effort or involvement based on the inferred measurements. As a first attempt, we conduct the analysis of interactions in each participatory forum separately. Therefore, we built a graph with vertices representing users and edges representing that two users had some relation (e.g., they had exchanged messages, or built a physical link between their wireless nodes)[1]. Then, we will study the community structure in each participatory forum separately.

Community structure is a common characteristic shown by most complex networks, which allows us to discuss common properties among their members. We analyzed the existence or not of community structures in our example as a result of interactions between their members. Each boxplot in the Figure 7.3 summarizes the nodes composition of communities detected in our participatory forums — Comm., Dev. and Users refer to the communication, dev-list and users-list graphs — using the clique percolation and Louvain methods (See 4.2.5). Members of a layer which do not belong to any community are not represented.

The structural differences observed in the formation of both types of communities is a consequence of measuring the participation of users individually on very different activities. In the main activity, for example, the amount of users belonging to one or more communities is too small to be used as part of the measurement. Users do not form strong bonds with their peers, because these interactions mostly occur during new and sporadic setups of hardware devices.

The interactions on supporting forums — mailing lists in this case — have higher chances to last, and hence form strong bonds among the members of the community. As an example, the *Louvain method* detects communities representing the 14.4% and 7.11%, of the users, while the *clique percolation method* reveals another community structure, enclosed by a core group of

[1] How to build these graphs from the interactions in a mailing-list is the topic of Chapter 8.

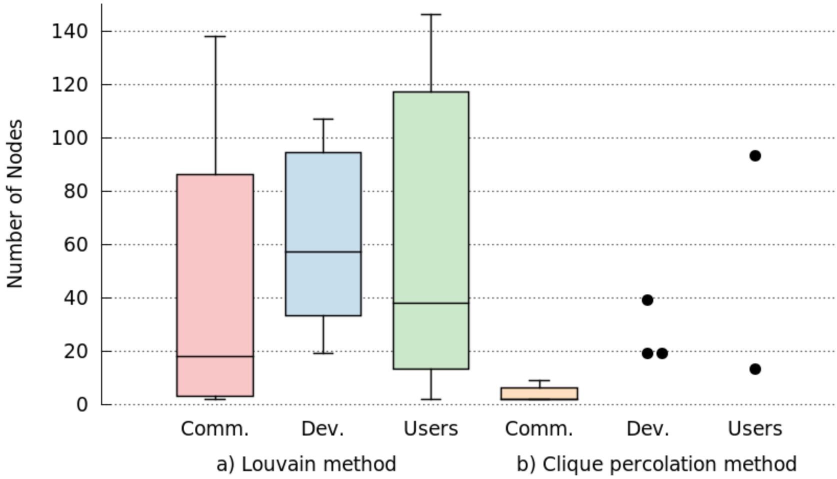


Figure 7.3: Number of nodes by community.

members representing around the 86.24% of the participants in both supporting activities.

We can conclude, hence, that **communities' structures either do not reveal clear patterns of participation in supporting activities or are too inclusive to properly distinguish between different relational patterns (I)**. Most probably, the social value of a community member in the common-pool resource is more influenced by its connectivity and position towards the rest of the participants than on his or her individual contributions. For instance, if somebody sends a lot of messages to a single person, it does not imply that he or she is generating any social value.

As an alternative, we measured the individual impact of the users inside the network as their closeness centrality, HITS authorities and hubs (See 4.2.6). Table 7.1. It revealed that there are only two users in common on all layers among the 10 higher ranked members, which are identified as the network founders. There are also two members in common in the mailing list which do not appear as top ranked in the main activity layer.

This is consistent with several studies about the structure of community networks [R2, R4] where their members reported that only a small fraction

Table 7.1: Top leaders in multiple participatory forums

user-id	Ranking schema	communication	users-list	dev-list
0	closeness coefficient	X	X	X
0	HITS authorities	X	X	X
1	HITS authorities	X	X	X
126	closeness coefficient		X	X
126	HITS authorities		X	X
32	closeness coefficient		X	X

of participants are interested on contributing to several facets of the resource-pool ecosystem. Usually, senior members well connected and integrated, who coordinate the different participatory forums. Thereby, **the participation of users in different supporting activities must be analyzed as a single and complex structure (II)**.

Roles and positional analysis might offer techniques to understand and group users according their influence in several graphs (solving problem I). However, as we show in Section 7.4 traditional methods are limited to measures of similarity based on pairwise comparisons, limiting the analysis to multi-layer or multi-relational structures (which are necessary to solve problem II). Thereby, the remaining **chapter focus on describing a new framework to find positions and roles using comparisons between actors and sets of actors instead of just using pairwise comparisons**.

7.3 Related work

7.3.1 Other related techniques to blockmodeling

Some recent advancements on generalized blockmodeling [16] are related with the problem of finding roles — or positions — based on different relations rather than direct connectivity of actors. In [110] authors use machine learning techniques based on latent links detection to infer the possible features defining the actors in the network, and hence to cluster actors based on them. As most direct approaches, the technique finds optimal or sub-optimal roles and clusters. However, as a notion of equivalence is not explicitly defined, the semantics of the identified blocks and which types of roles they represent requires an additional interpretation.

The work of Doreian et.al. [111] instead, proposes a mechanism to find roles in two-mode networks that is similar to the algorithm used in our proposal to find generalized roles in a matrix with different dimensions. The solution is efficient and effective when the two modes are disjoint sets of information, but it has not been defined for matrices relating actors with sets of actors. Finally, to our knowledge, none of the proposals in the literature of generalized blockmodeling differentiate roles and positions as two structures under the same assumptions of equivalence.

Several generalizations have been developed to find positions without perfect similarity/dissimilarity [112], to be used in weighted graphs [16], or even to find non-trivial equivalent positions [113]. While these approaches have proved useful to detect some kinds of positions, and are flexible enough to accommodate different kinds of similarity functions, they are also based on pairwise relationships. However, the general idea of finding approximate equivalences is also fundamental in our framework, because a strict check for equivalence would rarely identify any groups of similar actors in real social networks.

In this work, instead of just using pairwise relations we present a new framework to find positions and roles using *comparisons between actors and sets of actors*. With this change of perception, it is possible to find more complex positions, or to study traditional ones in more complex network representations.

7.3.2 Positions and roles in multi-dimensional graphs

A recent work published by Rossi and Ahmed [114] is the closest work to our idea of extended relations. In their proposal, the authors describe a new taxonomy for role discovery methods, which also introduces the idea of "feature-based role discovery". According to their proposal, the similarity between nodes can be measured using a set of node-structural features (e.g. degree, distance, etc.), which can be any set of measures taken from the initial graph. Together, they create a new matrix containing all the measures related to the actors.

Then, they use machine learning techniques to infer the social feature-based roles.

It is possible to argue that some of the extended equivalences proposed in our work could be used as features in their model, but in our framework we keep track of the relation between the measure — or feature — and the nodes related to it — the subsets of nodes that are needed to compute the measure. Because of this, our framework is able to measure not only patterns of relations (roles), but also positions.

7.4 Social positions and roles revisited

7.4.1 Roles and positions as different structural concepts

Position and role analysis has been used to explain several phenomena in social media. It has been used, for example, to categorize Wikipedia's participants into four groups — substantive experts, technical editors, counter vandalism editors and social networking editors — based on their contributions and interactions [115], with the aim of discriminating between editors with clear expertise and regular contributors. In [116], instead, authors examined posts and replies in Reddit to build a classifier to automatically detect the answer-person roles. The proposed classifier can be used, among other things, to understand which patterns encourage people to answer others and, hence, to build reward systems to increase the number of actors with this role.

Both concepts, role and position, have been redefined many times in the literature, both by mathematicians and sociologists, more or less in detail. In this thesis we lie on the definition provided by Wasserman and Faust [15]:

> *In social network analysis* ***position*** *refers to a collection of individuals who are similarly embedded in networks of relations, while* ***role*** *refers to the pattern of relations which obtain between actors or between positions. The notion of position thus refers to a collection of actors who are similar in social activity, ties, or interactions, with respect to actors in other positions.*

Despite the lack of mathematical notation, this definition clearly states the idea of identifying positions as a clustering problem where actors — vertices of a graph — are assigned to smaller subsets — called positions — based on a notion of similarity. It is important to note that this similarity notion not only measures how similar the local connectivity between pairs of actors is in the graph, but can also measure other properties of the vertices — called relations, or ties.

7.4.2 Equivalence as a similarity measure

Structural equivalence is the most basic and strict notion of similarity. Other similarities have been developed lately to relax the notion of equivalence. In *regular equivalence* [15], for example, two actors are in the same position if they have similar relations with other positions; while in *stochastic equivalence* [112] two actors are in the same position if they have the same probability distribution of ties with other actors, which is more similar to the notion of role that we are presenting in this work. A complete mathematical definition of all these notions of equivalence, and other variants, can be found in [15].

While the concept of similarity has always been tied to the concept of equivalence — local connectivity —, some alternatives have been proposed to include other relations in order to measure the similarity between two actors. These alternatives compute node-based or distance-based features, and use them as an extra constraint to evaluate node equivalence. It is possible, hence, to generalize the notion of *structural equivalence* from the original definition.

Let $G = (U, E)$ be a graph representing a social network, where U is a set of nodes representing actors and $E(i, j) = 1$ if nodes i and j are connected, 0 otherwise. Then, we can say that two nodes i and j are structurally equivalent (and so in the same position) if and only if [117]:

$$E(i, k) = E(j, k) \quad \forall k \in U; k \neq i, j \tag{7.1a}$$

$$A_h(i) = A_h(j) \quad \forall h \in H \tag{7.1b}$$

$$D_p(i, k) = D_p(j, k) \quad \forall p \in P; k \neq i, j \tag{7.1c}$$

where A is a set of H node attributes, where each node attribute is denoted by A_h with $h \in [1, H]$; and D is a set of P comparison functions, where each function is denoted by D_p and $p \in [1, P]$. Regular equivalence would further allow replacing k with previously identified positions, so that multiple iterations can be computed.

Notice that these relations still constrain the model to a) the adjacency connectivity matrix U and b) the pairwise actor comparison (i, j). For an extended taxonomy and classification we suggest the recent work of Rossi and Ahmed [114].

7.4.3 Roles and positions in complex networks

In the context of this thesis we need to identify social roles and positions in a complex network, product of the aggregation of different graphs representing the interactions among members of the commons social venues. Although the traditional methods (Section 7.3) were developed to detect positions also in multi-relational networks, the flatten or aggregation process cause a loss of information that could misguide the results.

To illustrate the problem, consider the social network represented in Figure 7.4, which shows the relationships between a group of eight actors. The network is represented as a multi-relational graph, where each relation — or layer — represents a friendship or a working relation between pairs of actors.

The analysis of the graph structure of each layer independently reveals social positions of interest only for the relationship under analysis. As an example, under the assumption of structural equivalence, we would like to group actors that are connected to exactly the same other actors in the layer. The positional analysis using the structural equivalence definition finds four positions, identified by color, in the *Co-workers* graph. They are resumed in Table 7.2. Using the same definition of equivalence on the *Friends* layer, the positional analysis will identify five different positions. Notice that while positions ρ_1 (blue) and ρ_2 (red) are exactly the same, members on positions ρ_3, ρ_4 and ρ_5 are rearranged differently on each layer.

The alternative, will be to flatten the graph into a single-relational graph with 8 actors and 11 edges before perform the structural analysis. In this case, the same analysis will continue placing actors A, B and C in the same

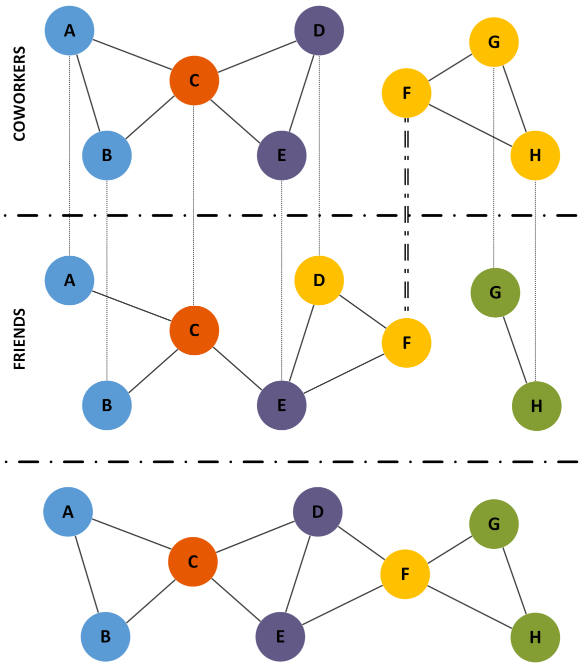


Figure 7.4: Complex example

Table 7.2: Structural equivalence positions in the multi-relational network example

Layer	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5
Co-workers	$A,\ B$	C	$D,\ E$	$F,\ G,\ H$	-
Friends	$A,\ B$	C	E	$D,\ F$	$G,\ H$
Flattered	$A,\ B$	C	$D,\ E$	F	$G,\ H$

previous positions ρ_1 and ρ_2, but it will still identify the subset of actors $\{F, D, E, G, H\}$ on different positions of interest. Even if the flattering process is very simple, the positions found in the new graph are a combination of the positions detected in each of the layers independently.

Both previous analysis turned out to be useful to discuss the structure of the social network, but none of them give us information about how the positions between layers are related and which effect they have in the structure of the multi-relational graph. Consider, as an example, the position of actors C and F in the Figure 7.4. From the point of view of the structure of each layer, the actor F is crucial to allow nodes $\{A, B, C, D, E\}$ communicate with nodes $\{G, H\}$ while the node C is certainly not.

7.5 Extended blockmodeling framework

Blockmodeling [15] is, to our knowledge, the most used and explored technique to detect roles and positions in social networks and, more generally, in any system that can be modeled mathematically using a graph. In blockmodeling, actors are grouped into *positons* — called blocks, sometimes *roles* — based on a similarity or dissimilarity measure between them. To compute this measure, actors are compared based on their social behaviour and structural connectivity in the network. In its original form, the similarity measure corresponds to the correlation between columns in the graph adjacency matrix, which results in including actors connected to the same other actors into the same position — as for the colored nodes in Figure 7.4.

In this section we describe our *framework for group relations* (Figure 7.8b) which allows us to apply blockmodeling analysis to find social roles and positions based on the global structure of the network, rather than being constrained to pairwise comparisons (Figure 7.8a). The framework has two basic components: the *extended comparison function* for group measures and the *computing algorithm* for identifying positions and roles:

- The *extended comparison function* is a two-dimensional matrix (M) that stores the similarity or dissimilarity between actors (rows) and sets of actors (columns). We first need to identify subsets of actors depending on the analysis we want to perform, then we must compute the comparison function.

- The *computing algorithm* used in our experiments is a generalization of the REGE/A algorithm proposed in [118] for regular equivalence. In general, any clustering algorithm already used in unsupervised blockmodeling analysis could be used instead.

7.5.1 Extended comparison function for simple graphs

To be able to compare actors with subsets of actors we need to replace the adjacency matrix typically used as an input for traditional blockmodeling with a more complex structure, capable of storing extended relations. This matrix will be used as input data for the computing algorithm.

Building it involves two main steps: a) dividing the actors into groups of interest and b) defining the comparison measure. These two definitions are interdependent, and need to be specified together.

As we have mentioned, our approach is based on a generic comparison function D. Let $G = (U, E)$ be a simple graph. Then[2],

$$D : (U, S) \to \mathbb{R} \tag{7.2}$$

where $S \subseteq 2^U$ depends on D.

We can then use the function D to build our extended matrix M, by substituting the adjacency matrix equivalence in Eq. 7.1a by our formula. Hence, we can define the extended matrix M as:

$$M(i, S_j) = D(i, S_j) \tag{7.3}$$

As a concrete example, consider Figure 7.5, showing Padgett's marriage network [51] with each family colored according to its approximate (which we will define later) social position (defined as *being part of the shortest path connecting pairs of nodes*).

Families *Albizzi* and *Guadagni* are connected to totally different nodes, that are themselves in different positions. So, they would not be considered being part of the same position by existing methods. For example, if we check the *Medici* family, *Albizzi* is connected to it while *Guadagni* is not. If we check *Lambertes*, *Albizzi* is not connected to it while *Guadagni* is. However, if we now consider the pair {*Lambertes*, *Medici*} and a comparison function:

[2] 2^U indicates the power set of U

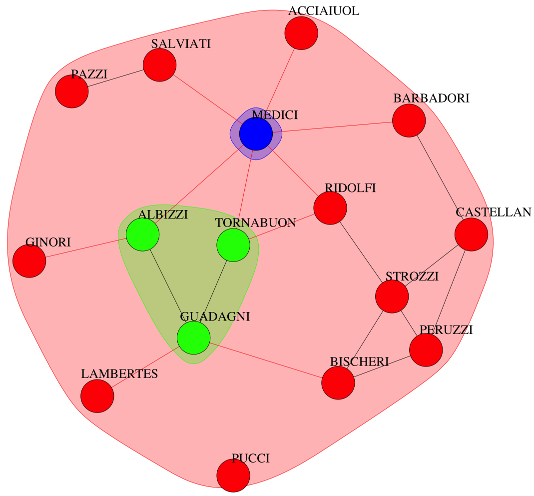


Figure 7.5: Padgett's marriage family network and approximate positions.

$$D(i, \{k, q\}) = \begin{cases} 1 & \text{if } i \in \text{short. path betw. k and q} \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

we can see that both *Albizzi* and *Guadagni* are on a shortest path between them. If we check other pairs of nodes, we can see that this is true in several other cases (e.g., to efficiently go from *Bischeri* to *Ginori* we should also pass through both *Albizzi* and *Guadagni*). In summary, *Albizzi* and *Guadagni* are included according our framework and the new similarity function D in the same position because they share the same relationship with other *pairs of nodes*, instead of single individuals.

Figure 7.6 shows the corresponding extended matrix M. Each cell in the matrix corresponds to one binary relation between an actor i and a set of two other actors (k, q). The rows and columns have been arranged in order to group together similar positional actors.

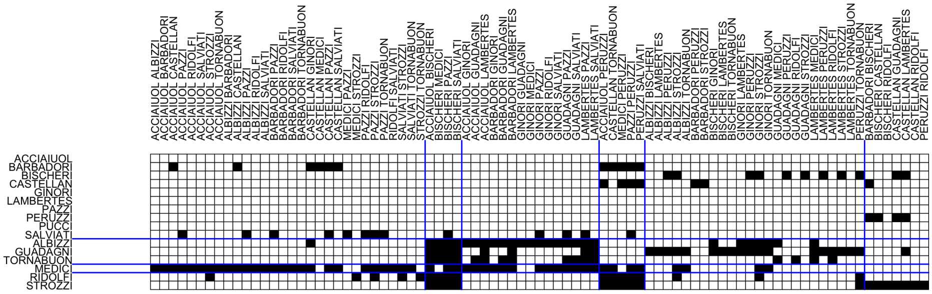


Figure 7.6: Extended relations matrix for the Padgett's marriage family network.

7.5.2 Positions and role assignment for simple graphs

After computing our extended comparison matrix, which associates each actor in the network with the subsets of actors used to measure the equivalence, the next step consists in partitioning the actors into $\beta_M = \beta_1, \beta_2, \ldots \beta_m$ similar positions and $\rho_Z = \rho_1, \rho_2, \rho_3, \ldots \rho_z$ similar roles.

While in traditional blockmodeling, only one of the partitions are calculated because both – roles and positions – are considered interchangeable concepts, determined uniquely by the definition of equivalence used; in our framework they are different concepts that can be measured from the extended relations matrix.

Positions are computed by clustering the rows of the extended matrix and forming groups of actors whose relation with the same subsets are similar. In the literature there are many clustering algorithms that can be used to make the rows clustering assignment. In order to simplify the results, and for comparison purposes with other indirect blockmodeling methods, we decided to use hierarchical clustering. In concrete, we first generate a dissimilarity matrix by computing the euclidean distance between rows of our extended matrix M and then we generate a hierarchical clustering using the Ward [119] cluster similarity function.

The same procedure can be also repeated for the columns in order to highlight to which subsets each block is related with — even though positions are only identified as subsets of the rows given the asymmetry of the matrix. Finding the optimal number of clusters for the positions assignments is discussed in detail in Section 7.5.3.

As an example of positional analysis, we used our method and depicted in Figure 7.5 the resulting assignment of actors that *are being part of the shortest path that connects pairs of nodes* into four positions. Table 7.3, instead, shows the corresponding roles detected using the same equivalence.

Notice that the two assignments are not identical. While the *Medici* family, as an example, is the single member of a role and a position, *Strozzi* and *Tornabuon*, who play the same social role, are in fact in different positions —- because they are in the same number of shortest paths, but between different sets of actors. Under a strict check of equivalences between the rows of the extended matrix, positions would be finer partitions of the roles. However, this is not guaranteed when approximate clustering algorithms are used.

Table 7.3: Roles identified in Padgett's marriage families network.

ρ_1	ρ_2	ρ_3	ρ_4	ρ_5
ACCIAIUOL	*CASTELLAN*	*BARBADORI*	*ALBIZZI*	*MEDICI*
GINORI	*PERUZZI*	*BISCHERI*	*GUADAGNI*	
LAMBERTES		*SALVIATI*		
PAZZI		*TORNABUON*		
PUCCI		*RIDOLFI*		
		STROZZI		
0	**4, 7**	**11 - 17**	**26, 27**	**50**

A similar procedure is followed to identify roles, using the same extended matrix M previously used to compute positions. However, instead of comparing their relations with each subset of actors, we use their patterns of relations — that is, some summary of the distribution of row values. Then, actors are grouped using any clustering algorithm.

7.5.3 Positions and roles uncertainty

We have previously mentioned the possibility of using different degrees of freedom for each definition of equivalence, especially when the notion of structural equivalence is used.

This is common practice in the blockmodeling literature: it is unlikely to find any meaningful structurally equivalent positions in networks with dense structures, like Padgett's marriage families network (Figure 7.5): the normal variability in connectivity prevents us from finding two nodes with many connections and connected with the exact same other nodes. As a result, every single actor in the network will be placed in a different position with only one member.

To relax the definition, and find meaningful positions and roles using indirect approaches, it is sometimes necessary to utilize some knowledge about the social network under analysis, e.g., specifying the number of expected positions.

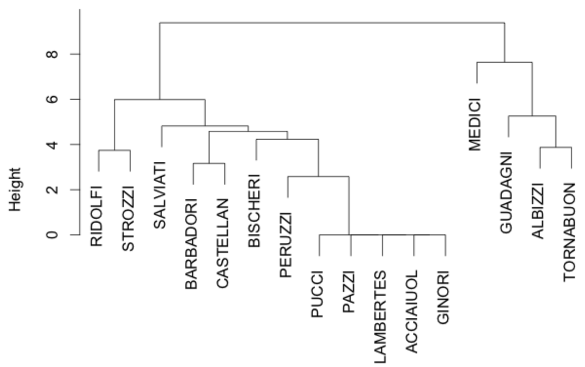


Figure 7.7: Dendrogram representing the hierarchical clustering in Figure 7.6

As an alternative, we can measure how good different partitioning are by measuring the relative distance between the maximum height of the hierarchical clustering and the height of the cutting point in the dendrogram. In Figure 7.7 we show it for our working example. Ideally, cutting the dendrogram at height $h = 0\%$ we will find positions where all the blocks are either complete or zero. As we pull up the cutting point we will increase the uncertainty, but we will be able to group actors into fewer positions. This relaxation of the block formation can be applied to all extended measures, and make them comparable in terms of uncertainty.

In our experiments we refer to "minimum height (h_{min})" as the minimum percentage of h needed, for a particular clustering, to find at least one position with more than one actor. Having a higher or lower h_{min} does not imply that a particular solution is more or less correct, but we can make the hypothesis that if larger positions are present in the data and the adopted similarity measure is appropriate, these will be identified with less need for approximations — on the other side of the scale, with $h = 100\%$ uncertainty all actors would be included into one single position.

7.5.4 Extending the framework for complex graphs

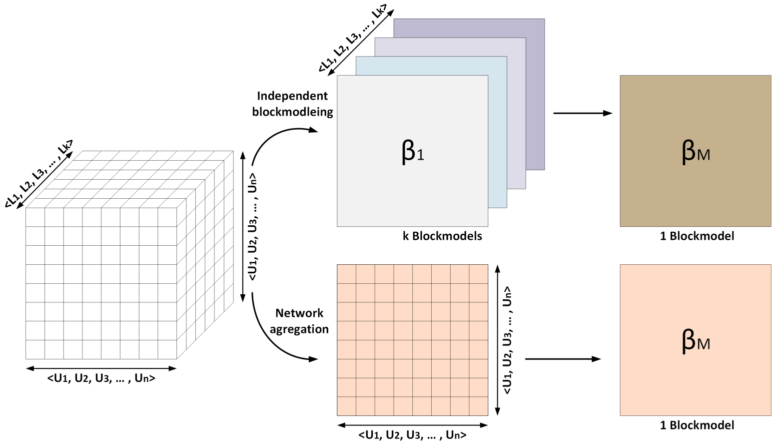
Following our initial motivation of detecting roles and positions in multi-dimensional structures, we need to extend the proposed methodology to capture this information. It requires incorporating multi-relational measures into the extended relations matrix (e.g., multi-relational versions of betweenness centrality) considering the cost of moving from one relation to the other.

Figure 7.8 allows us to compare the traditional blockmodeling solutions explored early (7.8a) with the extended framework for complex graphs (7.8a).

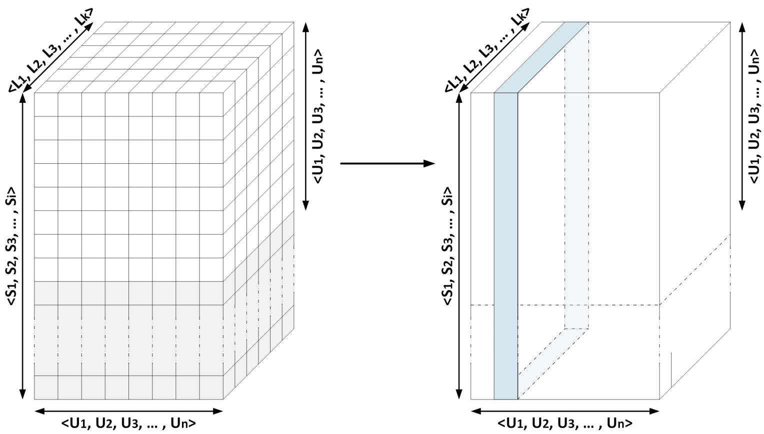
While the original matrix is a regular cube storing the k adjacency matrices of the complex network, the new structure is a non-regular structure where each cell represents the similarity value between a user i and a set S_j in layer k based on a specific comparison function D.

Then, instead of calculating positions individually, or using an aggregation matrix; in our framework they are computed using a multi-relational measure.

As an example, we could reuse our comparison function 7.4 by simply redefining the pair $\{k, q\}$ as *pairs of nodes not part of any component in any layer*. Therefore, the comparison function $D(i, \{k, q\})$ will represent if a particular node i is a *bridge* vertex connecting two disconnected vertices.



(a) Traditional blockmodeling strategies



(b) Multi-relational framework

Figure 7.8: Blockmodeling frameworks for multi-relational graphs

7.6 Results

In order to evaluate our framework we built a library in R using the *blockmodeling-package* [120] as a baseline. The library has been used to perform the experimental analysis, which included the detection of roles and positions with different combinations of simple graphs and equivalences. It simplifies the comparison of the identified structures with other equivalences well established, assuming that the extension of the methodology to higher-dimensional structures will still be meaningful.

7.6.1 Datasets

We evaluated our proposal using a set of real social networks, which are representative of the social structures that arise from the physical interactions among actors — like the Padgett's dataset — and are well known in the literature. Additionally, we created a set of synthetic networks for comparison purposes.

- **Florentine families** [51]: contains a two-relational graph describing the social relations among Renaissance Florentine families (person aggregates) collected by John Padgett from historical documents. Both relations — business ties and marriage alliances — were used as individual graphs.
- **AUCS** [121]: contains a five-relational graph describing the social relations among employees of a Computer Science department. The five relations are: lunch, work, co-authorship, leisure and Facebook friendship. For the analysis we have flattened the network into a mono-relational graph.
- **Synthetic data**: contains a set of networks built using the Erdos-Renyi [47] and Barabasi [50] models, with the same number of nodes as our real networks, but varying densities.

These networks are good representations of random graphs and the scale-free model which has been used for comparison purposes.

7.6.2 Other extended relations as equivalences

On previous sections we have described our framework using the example measure of **being in the shortest path (BSP)**, which intends to capture actors in positions of connectivity between the same other actors. Next we describe a set of easily interpretable measures, intended to demonstrate the possibilities of our framework, and how to build their corresponding matrix M.

- **Clique connectivity (CC)**: given the set of cliques in the network, the CC groups together actors with the same number of ties to the same cliques. Hence, the — valued — matrix M contains network cliques as columns and as values the sum of ties between each actor U_i and the members in each clique S_l.

- **Minimum Clique connectivity (MCC)**: given the set of cliques in the network, the MCC groups together actors with at least one tie to the same cliques. Hence, the — binary — matrix M contains network cliques as columns and 1 if the row actor has at least one tie with some member of the clique, 0 otherwise.

- **Community connectivity (COMC)**: groups together actors that are connected to the same maximum-modularity communities. Hence, the — valued — matrix M contains the communities as columns, and the percentage of ties to their members as a values.

- **Others community connectivity (OCOMC)**: groups together actors that are connected to other communities. Hence, the valued matrix M contains the communities as a columns, and the percentage of ties to their members as a values, discarding ties to its own community.

7.6.3 Framework validation

As any indirect blockmodeling methodology, the meaning and profitability of the findings reported by our framework are highly tied with the similarity measure used, which has to be chosen carefully. Potentially, any measure based on the topological structure of the social network — like distance-based measures — could be used as an equivalence.

However, in practice, it is necessary to discard measures of equivalence that do not find any dissimilarity — meaning, all actors are always grouped together — or measures where the assignment of actors into either roles or positions is a consequence of some random phenomenon. While the first problem is more dependent on the network structure (e.g., by definition we cannot find more than one position or role in Torus networks), the later is mostly related with our framework and hence we need to address it.

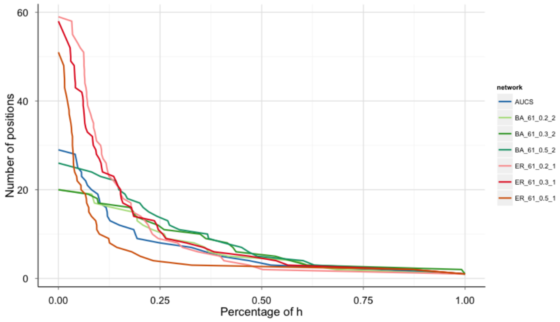
In order to verify the lack of randomness in our equivalences we tested the framework by comparing the positions found in our real social network with those found in 6 different synthetic random graphs. Figure 7.9 shows the number of different positions found as the percentage of uncertainty (h) using the several extended similarity measures. Measures are grouped together by pattern: the first plot (7.9a) groups together results regarding communities extended similarities — COMC and OCOMC —, the second plot (7.9b) regarding distance-based measures and cliques and the third one (7.9b) plots measures regarding only the minimum clique connectivity (MCC).

On each plot, synthetic networks noted as "ER-p" are built according to the Erdos-Renyi model with probability p, while the networks noted as "BA-p" are scale-free graphs according to the Barabasi-Albert model with exponent p. Each experiment has been repeated 10 times in order to avoid random effects caused by the network formation.

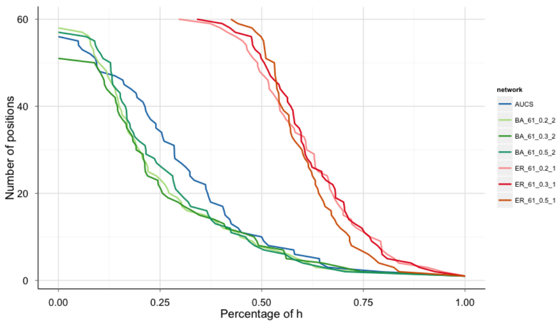
The experiment identified three different patterns of positions. Figure 7.9 shows, for each category, the number of positions found as *y-axis* and the corresponding uncertainty level h (*x-axis*). Hence, the left-most value of h for each curve represents the minimum uncertainty found in the measure (h_{min}). We can observe that in all three patterns the "real social" network always finds positions with equal (Figure 7.9a) or less (Figure 7.9b and Figure 7.9c) uncertainty as the synthetic graphs, and hence, closer to the ideal clustering of actors based on the definition of similarity.

In general, random graphs like the Erdos-Renyi networks present higher values of uncertainty h_{min} when our method is used. On the other hand, Barabasi networks are formed by a preferential attachment process, which is more similar to the interaction processes that form social networks, and hence is more likely that the positions found are capturing some social properties from the graph.

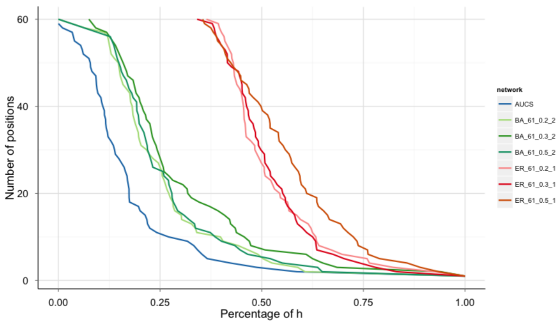
However, as the number of nodes in a graph decreases it is more likely that structural differences become less evident. As said, the same experiment, repeated with the Padgett's marriage network and the corresponding synthetic graphs with the same number of nodes reveal that all similarity measures have $h_{min} = 0\%$ and follow the same pattern shown in Figure 7.9a.



(a) Community based results



(b) SE, BSP and CC results



(c) MCC results

Figure 7.9: Number of different positions found as the percentage of uncertainty.

7.6.4 Similarity measures analysis

The different sizes of positions depicted in Figure 7.9a are related with the flexibility, in terms of measure, of each equivalence. More sub-settings of actors will increase the number of possible rows combinations, and hence, more heterogeneous positions; while fewer sub-settings will tend to create less positions with fewer uncertainty.

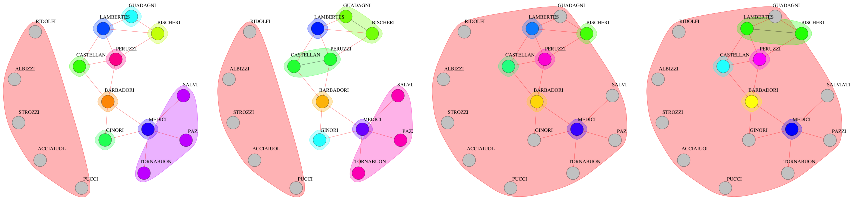
Another factor that may influence the flexibility of the measure is how the connectivity between actors and subsets is defined. As an example, for a given actor is not the same count if is connected or not to a subset of actors (e.g. a clique or a community), than counting if it has at least one tie with the subset, or the percentage of ties with the members of the subset. The measures proposed in this work, are good examples of meaningful measures for the analysis of social networks, but the framework does not constrain the definition of other interesting measures.

Another important aspect that must be taken into account for the interpretation of the results is the uncertain information management. The indirect blockmodeling methodology needs some guidance in order to distinguish between positions — or roles — related with the similarity measure, and other clusters containing actors not really captured. Consider, for example, our BSP extended measure, which tries to group together actors in the same shortest path as other actors. Then, by definition, nodes that are not in any shortest path between nodes or are completely disconnected to other nodes in the graph will be all placed in the same position by the clustering algorithm. However, they do not represent a position of interest, but rather a set of actors for which the measure does not apply.

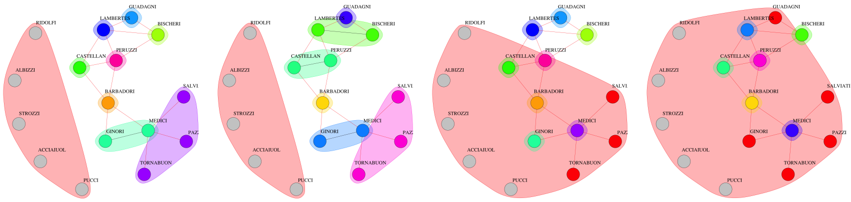
Figure 7.10 shows the first two positions detected — those with minimum h values — by our framework in the business social network of Florentine families. Each group of two graphs represents, for each extended measure, the clustering of actors into the two positions with smaller h. On each graph, nodes grouped together with the same color represent the positions detected on each case.

However, we marked with gray those nodes not considered in the measure – even if they are grouped in some position by the algorithm. These nodes are usually disconnected ones, leaves of the topology or — like *Ginori* family in the BSP measure — nodes connected in such a way that is not captured by the measure at hand.

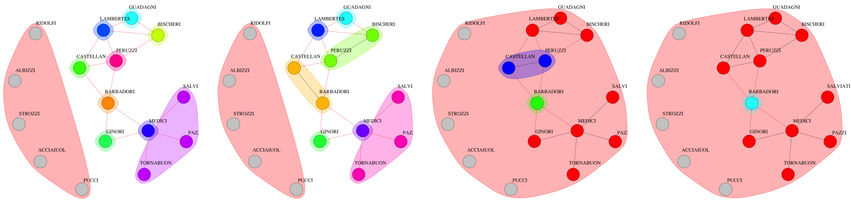
The individual analysis of the measures shows that, for the given social network, the framework is able to find positions with no uncertainty ($h_{min} = 0$), but their



(a) SE. h = 0% (b) SE. h = 25% (c) BSP. h = 0% (d) BSP. h = 17%



(e) CC. h = 0% (f) CC. h = 12% (g) MCC. h = 0% (h) MCC. h = 53%

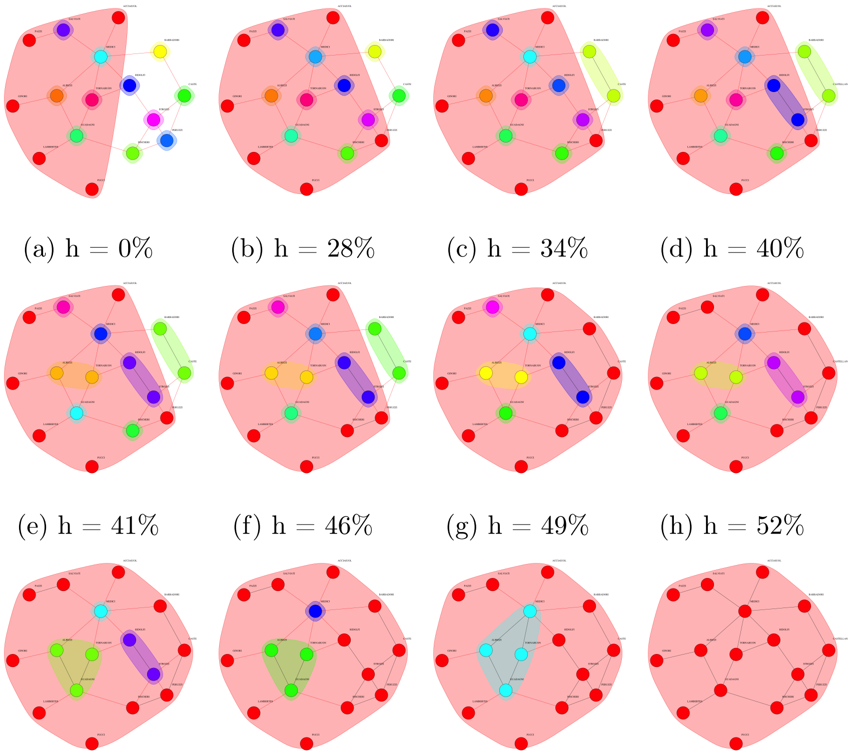


(i) COMC.h 0% (j) COMC.h 11% (k) OCOMC.h 0% (l) OCOMC.h 57%

Figure 7.10: Positions detected using traditional measures – structural equivalence (SE) – and the extended measures introduced in this paper for the Business social network of Florentine families, at varying levels of uncertainty

interpretability is not always clear. It occurs because the clustering algorithm groups together too few nodes. As an example, the OCOMC measure finds two clear positions which corresponds to the actors connecting the upper community with the bottom one; while the BSP measure identifies each actor with a single position of one element.

One possible solution is to analyze higher uncertainty positions (those with $h > 0$), but this requires human supervision to interpret the results. As an example, Figure 7.11 shows the all possible positions detected using our extended method for the Marriage social network of Florentine families, at varying levels of uncertainty using the BSP measure.



(a) h = 0% (b) h = 28% (c) h = 34% (d) h = 40%

(e) h = 41% (f) h = 46% (g) h = 49% (h) h = 52%

(i) h = 56% (j) h = 64% (k) h = 82% (l) h = 100%

Figure 7.11: Positions detected using the extended BSP measure at varying levels of uncertainty

We can see how actors are being grouped into different positions, as the level of uncertainty approximates the 100%. From the picture, we could be tempted to pick some result with uncertainty larger than 50% as the measure seems to be capturing some structure in the network. Nevertheless, this would be a mistake as we don't really know if the similarly measure chosen exists or not in the original network. Therefore, our only solution is to lie on the *minimum h*.

7.7 Conclusions

In this chapter we introduced the idea of *social effort*, as a measure of the participation of users in common social venues which are part of communities built around common-pool resources. This supporting activities are of great interest, because they are necessary to organize and coordinate the main activity (e.g., to guarantee the users' compliance with the common governing rules and mechanisms presented in the previous chapters).

Our first analysis shown that detecting and clustering users based on their participation would require the development of new mechanisms for measuring new types of social positions and roles in simple and complex models, like multi-relational networks. Therefore, we have proposed a conceptual extension of blockmodeling that allows us to plug in additional comparison functions not usable in a standard setting. At the same time, this extension enables the discovery of new kinds of positions also on simple networks: in this chapter we have focused on this aspect, providing an analysis of the new possibilities and limits of our proposal.

More in detail, according to our proposal, to enable the usage of the additional types of similarity functions discussed in this work it is necessary to change the regular similarity matrices for a more complex structure able to relate actors in a network with a) the extended measure and b) the extra information used to compute such measure – in this case, subsets of actors. These new measures generate a new asymmetric equivalence matrix, that can be analyzed to find both social roles and positions.

The selection of the extended measure depends entirely on the objective of the analysis and/or the meaning of the positions and roles desired. Theoretically, any measure computed in a graph as a result to apply a given function over a node and a set of nodes would be a candidate. In practice, we have observed that some of the measures generate higher values of uncertainty (h), that is, they require a higher level of approximation to identify positions containing multiple nodes. Therefore, the interpretation of the results will be more difficult.

Although during the analysis presented in this chapter we have used several new definitions of extended measures, they are provided as simple examples based on well-known concepts in networks. It is expected, hence, that community members agree which similarity measure to use based on the social venues they have, and the kind of roles they want to reward.

Part III

Discussion

Chapter 8

Community based incentives for cooperative applications

The mechanism schema introduced in this thesis uses the community of users as an externality to enforce that all the participants comply with the incentive (See Chapter 5). In Chapter 7 we have shown, however, that users are naturally and spontaneously inclined to participate in only one type of activity. In order to measure their participation on the supporting tasks, we have developed a new detection framework for identifying roles and positions in multiplex graphs, but there are still some questions unsolved in practice.

The objective of this chapter is to review all aspects of the schema not previously solved and provide a practical solution for each one of them. This chapter addresses, firstly, how to built a graph representation of users' interactions to perform the analysis of roles and positions announced early on. Then, we complete the incentive scheme by studying how influence the sharing outcome of users using the information provided by the supporting activities. Finally, we will generalize our schema for larger scenarios with multiple types of resource.

8.1 Introduction

The active participation of community members is key for the success of the proposed incentive mechanisms, since we assume that they will implicitly enforce users to comply with the rules designed. However, the analysis performed in Chapter 7 shows that in absence of some externality, users will not participate in both types of activity — main and supporting — simultaneously.

Users' participation in the cooperative process is stimulated explicitly by the mechanisms designed in Chapters 5. Therefore, we designed a new schema that rewards their participation in supporting activities giving them in return higher cooperation opportunities in the common-pool resources (See Figure 8.1).

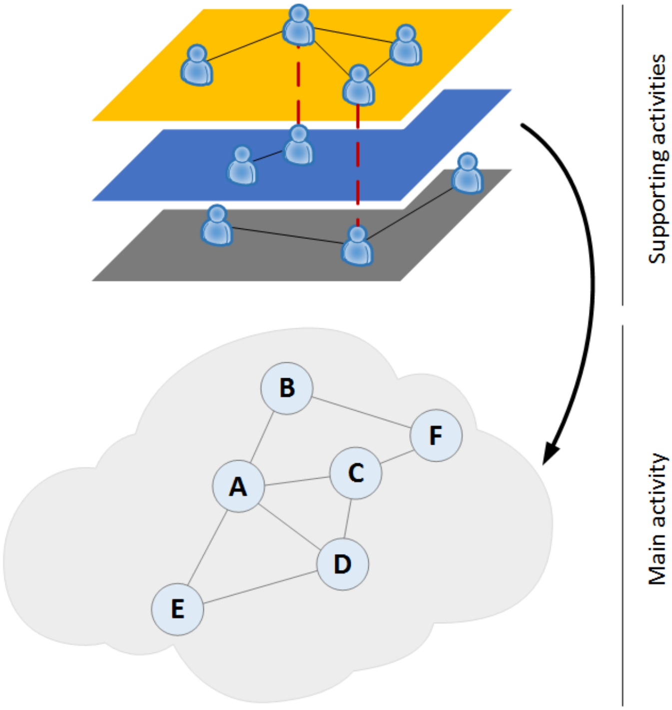


Figure 8.1: Incentive framework for cooperative applications

The top of Figure 8.1 describes the social component of the framework, which monitors the activity of users in several supporting forums, and reports a single "social effort" score based on their positions and roles using the methods described in Chapter 7 for simple and complex graph structures. Modeling all supporting activities as a graph is an open problem that each application must solve depending on several environmental factors (e.g., how users interact between them, what supporting activities they want to account).

As actually most of these interactions occur through online social forums, in Section 8.2 we discuss a method to model users' interactions in mailing-lists as a multiplex graph structure. In this section, our intention is to provide a useful example, rather than a generic method for all sort of participatory forums.

In Section 8.3 we finally introduce a simple model for transferring the social effort score to the main activity (See Figure 8.1), allowing users to either compensate the lack of resources shared with higher supporting tasks or to compete for larger sets of resources than other peers with the same sharing ratio.

Finally, in Section 8.4 we generalize our framework for applications with multiple shared resources. The new design allows system designers to score users' effort as a linear combination of several activities, and to dynamically weight the impact of each task by the type of resources it needs.

8.2 Graph generation from supporting activities

Users' participation — or effort — is a vague concept that can embrace almost any activity performed by community members. In the context of this thesis we decided, instead, to measure the results of such effort. Hence, assuming that users devoting similar effort and time to the supporting activities will have similar patterns of interactions, we decided to score users' participation by identifying their social roles and positions.

In this section we explore how to build the users' graph of relations, using the interactions happening in on-line communication platforms and, more specifically, on their community mailing-lists. The overall objective is to provide just an example of how to extract relationship information from on-line sources. This method can be used with other participatory forums (e.g., online discussion forums or live-chats) too, but it will require to be adjusted properly.

The method described in this section has been tested using Guifi.net [26] mailing-lists data. Guifi.net network is a community network (See Section 3.1.2) started in 2004, and in 2014 reached more than 24,000 operational devices, most of them in Catalonia and nearly all of them in the Iberian peninsula. The community has 42 mailing-lists for general purposes (e.g., discussing new deployments, helping new users).

8.2.1 Extracting relations

The objective of our schema is to transform the information contained on all the threads in a mailing-list, to multiple complex graph structures — one per mailing-list —, losing as few information as possible in the process. In this way, each of the structures can be simplified (e.g., by selecting some temporal windows, flattern the multi-relations) or grouped in a single multiplex graph later.

Figure 8.2 shows in the left-side a tree representation of two original threads in one of the mailing-lists studied. Each vertex in the tree represents a new message, and its label indicates the user who has sent the message. Each children of a given vertex represents a reply from some user to its parent message. All the subtrees of the original graph can be treated as a new thread conversation, or as part of the original thread. Our methodology is indifferent to both interpretations.

The right-side of the Figure 8.2 instead, shows the resulting complex graph structure after processing the two original threads. There are two types of vertex: users — represented as circles — and colored squares — representing an answer message from one of the two original threads. The label of each circle is a unique identifier indicating the user sending the message, while the label of each square is an incremental integer indicating the order — in practice, the timestamp — when each message has sent to the mailing-list. The edges in the right-side of the figure show the relation between users, threads and messages.

In Figure 8.3 we have separated both types of vertices — users and messages — and their relations, to simplify the understanding of their relations. Figure 8.3a shows the messages' graph, where we can appreciate that the red arrows indicate the order of the messages in the thread. As an example, the brown message 0 is the first message sent in the second thread — originally from user G. It received three answers, which are messages 1, 2 and 4. We can observe that

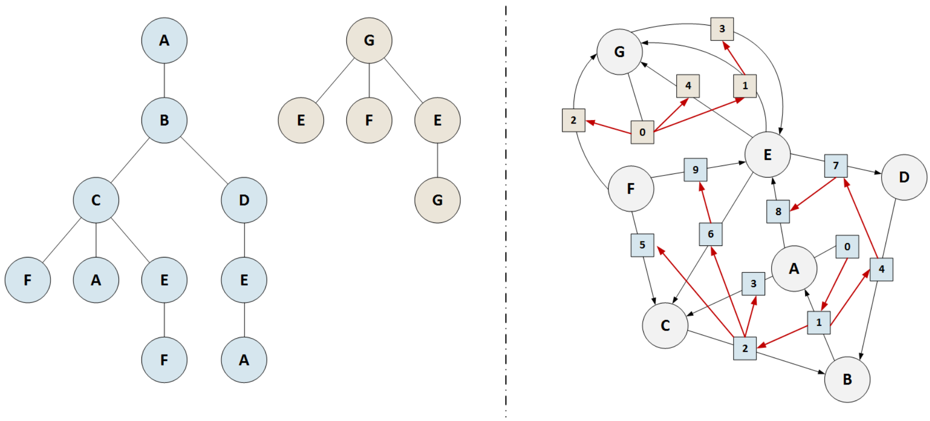
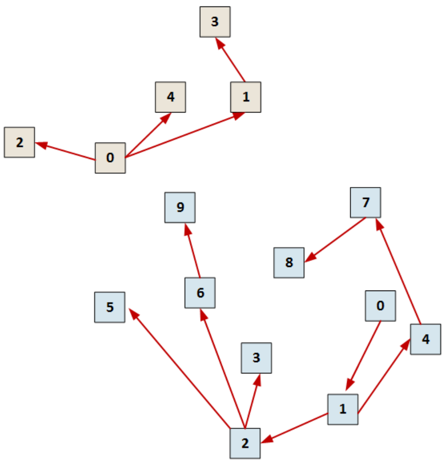
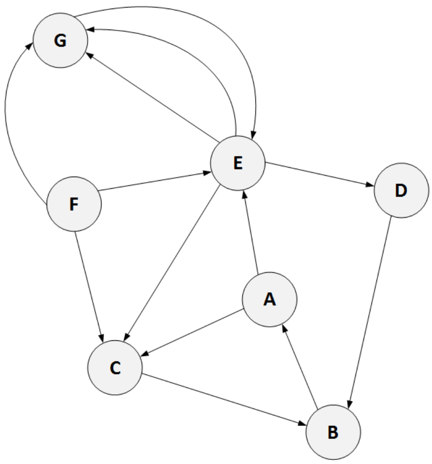


Figure 8.2: Graph generation process

this structure maintains both, the logical and temporal order of the information in the online supporting forum. Figure 8.3b instead, shows the users' graph with vertices representing users and each edge (i, j) an answer from user i to a previous published message from j.



(a) Messages' graph



(b) Multi-relational users' graph

Figure 8.3: Resulting networks of relations

The main problem with the last representation is that it does not provide any information about who started or finished each of the possible conversations in the original mailing-lists. For example, we can appreciate in Figure 8.2 an undirected edge between user G and the brown message 0 — which does not have any other edge — indicating that G was the starter of the original thread. Additionally, the messages' graph might provide extra information about the intention of the message as each square node stores the entire message information (timestamp, subject, text and attachments).

In practice, for the structural analysis we described in Chapter 7 we used the users' graph, grouping all edges between two vertices in a single indirected one, with a weight equal to the number of original edges in both directions. Other designs and applications might get advantage from the information provided by the complex network graph (See Figure 8.2) for building their incentive schema.

8.2.2 Homonyms detection

Homonymity is a characteristic of most distributed systems, like Peer-to-Peer applications, which implies the existence of users with multiple identifiers in the network [122]. Usually, homonymity emerges naturally because the system does not provide a single authentication mechanism. For example, in most community networks users have to register themselves independently on each mailing-lists in which they want to participate, thus providing a different email address each time.

Detecting homonyms is necessary to relate the information from different sources. More importantly, homonyms detection can be used as a first step for detecting whitewashing [123] — users' changing of identity to avoid being evaluated by their past activities.

When authentication method is based on the users' email, we can take advantage of the fact that most of the email addresses are a combination of the name and surname of the real users and, hence, they will be similar. In our experiments homonymity was detected using the email similarity rule suggested by Bird et.al. [124], which is based on the Levenshtein edit distance between email address bases (the original without the domain information). Although this method provided good results in practice — only a 7% of the homonyms were not detected — it needs higher refinements to avoid human intervention.

8.3 A complete framework

In this section we propose a regulation mechanism, able to reward users, not only for their effort in contributing with physical resources, but also for their work in supporting activities that benefits the community of participants (e.g., based on the social effort score developed early on Chapter 7).

Using this regulation mechanism, participants in common-pool resources, like community networks (See Section 3.1.2), will be willing to contribute more often in supporting activities. Moreover, those who are already dealing with these tasks can get a higher reward when they need to use the community applications or infrastructure.

In order to evaluate the proposal, we conducted several simulations using an extended version of the n-players Iterated Prisoner's Dilemma (n-IPD) framework. We modified the payoff matrix of the n-IPD to include a small social reward, which represents a compensation for users involved in supporting activities.

8.3.1 Social rewarding model

The simulation model consists on two different networks, where nodes — representing users — can be present in one or both topologies, as it represents the Figure 8.4. The top network ($G_{supporting}$) represents the social structure of one supporting activity, or a flattern graph extracted from multiple ones. In this network we have split the nodes into two categories — representing two possible roles or positions —, being *active* with a random probability p and *passive* with a probability $(1-p)$. In the bottom network (G_{main}) nodes will play an extended version oft he n-player Iterated Prisoner's Dilemma, where each node i present in both networks and considered *active* in the $G_{supporting}$ graph will be rewarded with an extra payoff α_i if has decided to cooperate with its neighbours.

Table 8.1 shows the payoff matrix of our extended version of the n-IPD. Without loss of generality, we can fix the cooperation reward ($a = 1$), the punishment to defeat ($\varepsilon = 0.05$), and the value received for cooperating ($c = 0$). Thus, we can study the influence of the other parameters in the game, which would allow performing a simplified analysis of the results [125].

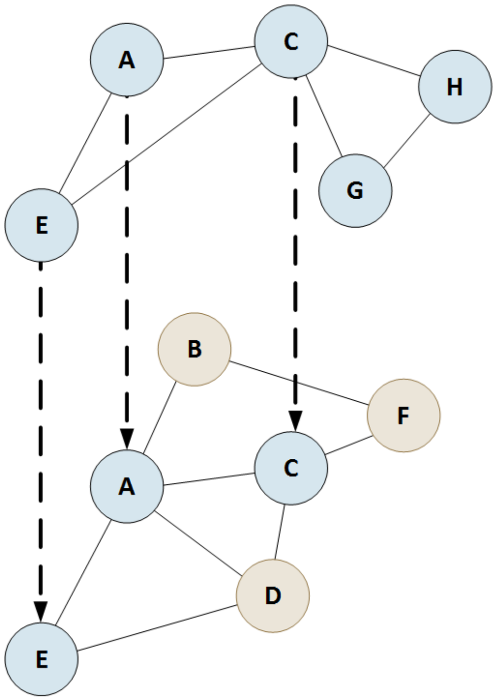


Figure 8.4: Participation's transfer model

Table 8.1: Payoff matrix of Extended Prisoner's Game

Player decision	**Co-player Cooperate**	**Co-player Deflection**
Cooperate	$(a+\alpha_1, a+\alpha_2)$	$(c+\alpha_1, b)$
Defection	$(b, c+\alpha_1)$	$(\varepsilon, \varepsilon) \to 0$

To prevail the dilemma in this game, the punishment parameter ε must take a value in the interval [0,1), and the payoffs rule — originally, $b > a > \varepsilon \to 0 > c$ — must be updated to $b > 1 + \alpha_i > \varepsilon \to 0 > c + \alpha_i$ for all i nodes. After the simplifications of the reward and punishment values, we can state that the α interval must be (0.005, 0.4). If this relation is preserved, the participants do not need knowing the value of their neighbors' social reward α_i to decide if cooperate or defeat, because such a value does not influence the game.

Therefore, the game could be simplified to a symmetric version, where the unknown information does not change the philosophical dilemma.

As our extended version of the game prevails the original Prisoner's dilemma spirit, it is well-known that the rational choice is always to defect, not only to prevent betrayals by other players, but also because it always gives higher payoff, no matter what the other player does (it is a Nash equilibrium [126]). However, it is also clear that the community welfare would be higher if both nodes decide to cooperate, and just here lays the dilemma. In order to address this challenge and understand the consequences of our proposal, we developed two extended versions of the prisoner's dilemma framework.

- **Extended Prisoner's Dilemma (EPD)**. This version corresponds exactly to the model described before, where for each interaction between two players, we add a social reward in case of cooperation.
- **Minimum Extended Prisoner's Dilemma (MEPD)**. In order to test the consequences of smaller changes into the original prisoner's dilemma game, we modified our extended payoff matrix. In this case the social reward is only added to each node once per round, resulting in smaller increments of the payoff for the cooperating nodes.

8.3.2 Modeling social dynamics

During the first round of the n-IPD game, all nodes make an individual decision between cooperate or defeat against their partners; these alternatives have equal probability to be chosen. During the next rounds, the system dynamic is controlled by the update strategy of the nodes (i.e., the nodes can change their cooperation strategy depending on the results of the past rounds). We evaluated our proposal using several update strategies, which are representative of the way that users make decisions in their real life [127].

- **Voter Model (VM)**. When users adopt this strategy, they behave similar to a particular neighbor. This imitation process has shown to be representative of many people during electoral processes [128], and it is a suitable manner to use social imitation mechanisms to deal with a strategy problem.

- **Unconditional Imitator (UI)**. This strategy is similar to the previous one, but the user imitates to the neighbor with the best payoff, provided that such value is larger than his own. The UI strategy might resemble the actions of the people in real life, when some neighbor has enough information to make suitable decisions [127, 129].

- **Replicator Rule (REP)**. With *REP* nodes choose a neighbor at random. Then, if the payoff of the chosen neighbor is larger than the node's own, the node adopt the neighbor's strategy with a probability proportional to the difference between both payoffs. In other case, nothing changes. *REP* is pairwise and stochastic; i.e., a node decides how to evolve, watching only one neighbor per round, and the result of the evolution is not univocally determined.

- **Unconditional Imitator (UI)**. In this case, a node select a neighbor with probability proportional to their payoff, without considering whether it is larger than its own or not [130]. MOR works as a local and stochastic process that allows the individuals to make mistakes; i.e., there is a non-zero probability to imitate a neighbor with bad fitness. This strategy can be considered as a weighted social imitation, where the weight is the success of the observed node.

8.3.3 Modeling social dynamics

The whole n-IPD game used for evaluating our proposal is described in Algorithm 5. For simplicity, the algorithm only includes the initialization stage and the main loop. The initialization stage has two main functions: **CalculateAlpha** function assigns a positive social score α to node i if, and only if, the node is in the set S. **calculateInitialAction** just assigns the first action of each node i as *Cooperate* or $Defeat$ with 50% probability each.

Then, the main loop starts and is executed for a fixed period of time $t = [0, T]$ (usually 250 rounds). On each of the executions, the algorithm calculates the payoff of nodes during the given round as the sum of the payoffs for each interaction of the game according the EPD payoff matrix (lines 7:10), and decides if nodes will change strategy by imitation or will continue playing the same strategy next round (lines 11:18). Functions **chooseCandidate** and **imitate** are, therefore unique for each imitation strategy.

Algorithm 5 Extended n-IPD game

Require: $G_{main}(V, E)$ ▷ Network graph
Require: S ▷ Set of active nodes in the supporting network
Require: $EPD(b, \alpha)$ ▷ Extended payoff matrix
Require: T ▷ Simulation time

for all $i \in V$ **do** ▷ Initialization
    $i.\alpha \leftarrow$ CALCULATEALPHA$(i, S, EPD.\alpha)$
    $i.action[t] \leftarrow$ CALCULATEINITIALACTION
end for

$t \leftarrow 0$
while $t \leq T$ **do** ▷ Main loop

    for all $(i, j) \in E$ **do** ▷ Payoff computing phase for t
        $i.score[t] \leftarrow i.score[t]+$CALCSCORE$(i.action[t], j.action[t], EPD)$
        $j.score[t] \leftarrow i.score[t]+$CALCSCORE$(j.action[t], i.action[t], EPD)$
    end for

    for all $i \in V$ **do** ▷ Imitation phase for t + 1
        $j \leftarrow$ CHOOSECANDIDATE$(i, i.neighbors)$
        if IMITATE$(i.payoff[t], j.payoff[t]) = true$ **then**
            $i.action[t+1] \leftarrow j.action[t+1]$
        else
            $i.action[t+1] \leftarrow i.action[t+1]$
        end if
    end for

    $t \leftarrow t + 1$
end while

return G_{main}

As an example, according the *REP* rule, nodes first choose a neighbor at random using the **chooseCandidate** function, and then decide if they will change or not strategy. Algorithm 6 shows, as an example, the **imitate** function of *REP* rule. Firstly, the function compares if the calculating player has more or less payoff than its opponent (line 1). If it has more payoff, then computes the differences of payoff as a probability (line 2) and tests it against a random number generated using a continuous uniform distribution (line 6).

Algorithm 6 Imitate function for REP strategy

Require: $PAYOFF_a$ ▷ Payoff player node
Require: $PAYOFF_b$ ▷ Payoff co-player node

1: **if** $PAYOFF_b > PAYOFF_a$ **then**
2: $\quad probImitate \leftarrow 1 - \frac{PAYOFF_b - PAYOFF_a}{PAYOFF_b + PAYOFF_a}$
3: **else**
4: $\quad$ **return** $false$
5: **end if**

6: **if** $probImitate > U(0,1)$ **then**
7: $\quad$ **return** $false$
8: **end if**

9: **return** $true$

8.3.4 Positive and negative effects on the community welfare

The most straightforward way to determine the effects of the proposed incentive model, is measuring the average of cooperating nodes density, as function of the parameter α, and then compare the results with the regular PD. Figure 8.5 shows the improvement of the *EPD* for two specific strategies (*REP* and *UI*), with a reward to betrayers $b = 1.1$ and 20% of users socially rewarded from 0.005 to 0.045. The improvement was calculated as the difference between the average cooperation during the last 250 rounds for each individual simulation. Positive values represent improvements produced by our proposal, while negative values represent the opposite.

In Figure 8.5 we can also see that when the community uses a *UI* strategy, it is not clear which of both frameworks — *PD* or *EPD* — is more beneficial for the social welfare of the community. However, when the users use a *REP* approach on the *EPD* framework, there is a higher probability of improving the cooperation inside the community.

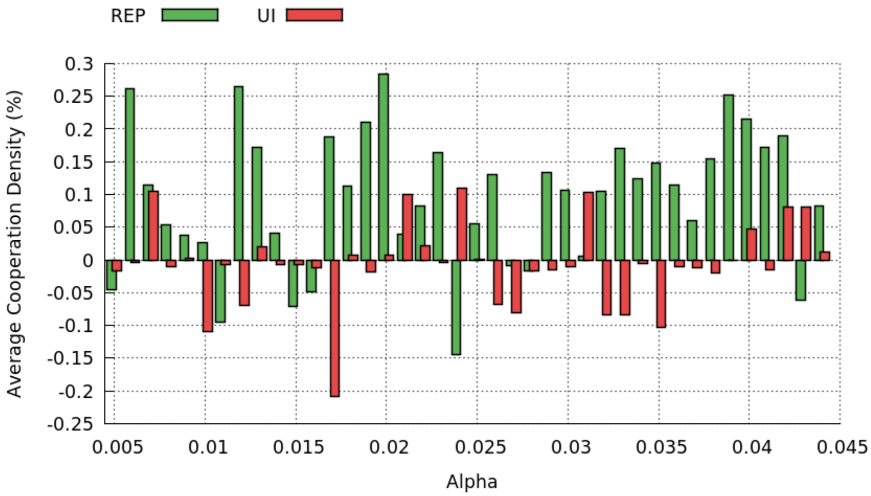


Figure 8.5: Average cooperation density using the *REP* and *UI* strategy.

The presented results are not enough to make strong conclusions about the existence of minimum values of α that ensure higher incentive using the *EPD* framework. However, Figure 8.5 shows that, as a general rule, higher values of α appear when the *EPD* framework is used. As an example, less than 9% of the values of α above 0.02, have generated a negative impact higher than 0.03.

Figure 8.6 helps us understand how each update strategy influences the cooperation on the three PD frameworks. Comparing the overall behavior of the system, we can state that *MOR* strategy does not allow the cooperating nodes to survive despite the used incentive mechanism. On the opposite side, the *VM* and *UI* update strategies are appropriate for reaching high percentages of cooperating nodes.

The minimum values of social reward were reached using the MEPD framework. This happens because the small changes introduced in the MEPD payoff matrix played a negative effect in the cooperation level of the system. Therefore, we can argue that when nodes compare their payoff after each round, the

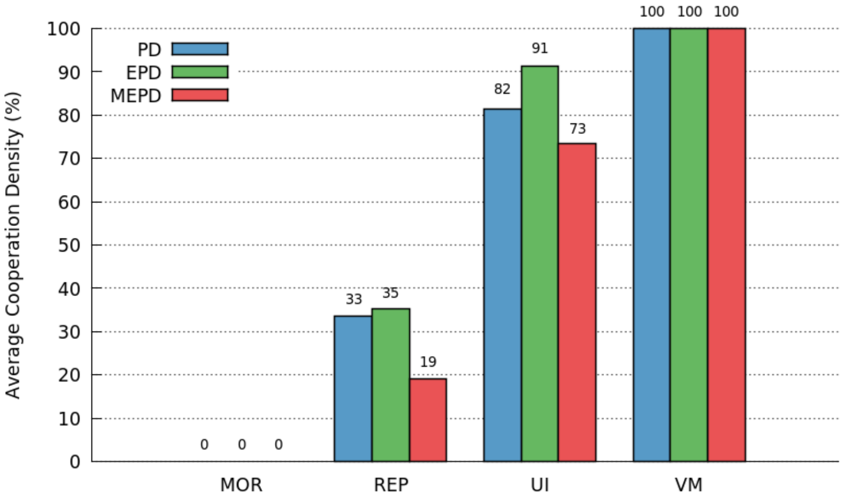


Figure 8.6: Average cooperation density using three PD frameworks with different update strategies.

contribution of the social reward to nodes' payoff is not enough to surpass the effect of the temptation reward, and hence, most of the socially rewarded nodes decide at some point to imitate some defeater neighbor.

The EPD payoff achieves about 11% of improvement, compared to the regular PD payoff value reached when users' update their cooperation decision using all the information about their neighbors (i.e., a *VM* strategy). However, the EPD framework does not affect the density of cooperating nodes when decisions are taken locally (i.e., when using *VM* and *REP* strategies).

8.3.5 Influence in presence of highly defection reward

Provided the previous results have been obtained for a specific combination of payoff values, it is important to check their generalization when the relationship between the payoffs changes. Figure 8.7 shows the average cooperation percentage achieved by the community for different simulations, using a VM update strategy, an EPD payoff matrix and a 60% of socially rewarded nodes. The results show that regardless the reward value (α), as the reward to the betrayers (b) increase from 1.1 to 1.4, the overall cooperation drops from

100% to almost 0%. However, it is also noticeable that higher values of social reward can help system designers to maintain the cooperation density between reasonable values.

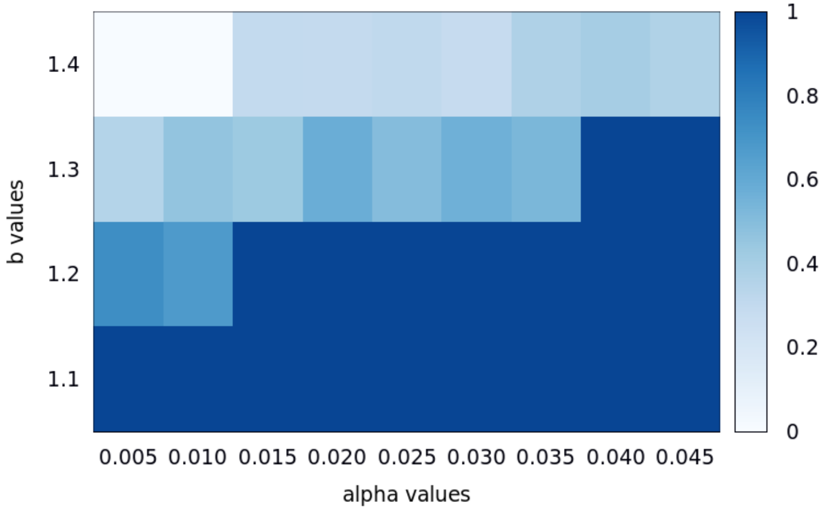


Figure 8.7: Average cooperation density of VM strategy, as a function of b and α

This result has a number of implications about the basic equation that we have assumed for the PD framework. For instance, and additionally of the ratio between the temptation value b and the punishment value ε, the designers have to carefully consider the implications of the α, which – in a limited way – can play a role similar to ε during the game. However, as the social reward only needs to be applied to a fraction of the community, it avoids increasing the payoff to mutual cooperators. Provided the social reward cannot exceed the value of ε, these strategies cannot deal with high values of b, like 1.4.

The relation between both, b and α has also been observed in other combinations of update strategies and percentages of nodes. However, sub-communities involving socially rewarded users need a higher α to counter-balance the increments of b.

8.3.6 Evaluation using real community data

It is apparently clear that rewards should be assigned to people contributing to the community welfare, even if they perform supporting activities. Through an exploratory study we extended the PD framework in order to consider a social reward for community members doing these activities.

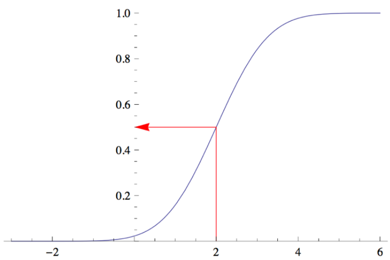
The first observations lead to the conclusion that even small values of social reward, have a direct impact (mostly positive) on the overall community cooperation level. It happens because (1) the risk to be defeated after a cooperation has also been reduced for some nodes, and (2) the reward to users that conduct a mutual cooperation has been increased. This consideration is consistent with the results reported in [131], where it is shown that individuals tend to react more positively to truly rewards on cooperation in the context of a public goods experiment.

Our study also found that there is a threshold ($\alpha = 0.02$) over which the difference in cooperation using the EPD framework is always positive, even in presence of high rewards to betrayers. The other parameters of the game, such as the users' update strategy and the percentage of rewarded nodes, can drastically modify the impact of α and b. Therefore, it is important to consider that the proposed mechanism only incentives the target community, when users' update their strategy using local information.

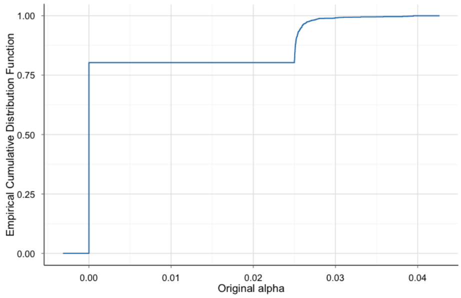
However, in Section 5 we shown that in cooperative applications — where users can freely choose their cooperation level — there is a wide range of cooperation ratios, in contrast to our early experiments. Similar situations cane be found in online social networks [132, 133]. Therefore, in order to enrich our results and to test the robustness of our proposal; we consider next a more realistic scenario based on real data gathered from users' behavior in Guifi.net [26], the largest active Wireless Community Network (See Section 3.1.2).

In this new model, we have substituted the random position assignment by a new similarity function, based on the number of messages sent by each user during the whole period of time. Then, we computed the Probability Integral Transformation (PTI) of the interactions between users, which maps the Cumulative Distribution Function (CDF) of a given continous random variable X to a uniform distribution [79], and removed all the values below a given percentile (q). As a result, we have a function that describes the distribution between the different social efforts done (measured as the number of messages sent) by the $(1-q)$ percentage of most active users in the network.

During the normalization process we have cropped the results to the range [0.025, 0.045], in which the α values guarantee a higher percentage of cooperation for the rewarded nodes. Figure 8.8 shows how probability functions can be normalize by mapping their Cumulative Distribution Function (8.8a), and the result after applying the mentioned technique to our dataset (8.8b).



(a) Theoretical model



(b) Real model (number of messages)

Figure 8.8: Mapping function from social effort-score to α

This method has two clear advantages: (1) it can be adapted to any probability distribution regardless the way used to compute the social effort score, making the new score comparable to any previous incentive and (2) it maintains a proportion of scores based in the original heterogeneity of the users' social effort.

Figure 8.9 shows the distribution of cooperation percentage achieved by the nodes, when some portion of them (p) is rewarded using the proposed motivation mechanism. Each line represents a group of users, considering $b = 1.1$, and a *REP* update strategy, to make the results comparable with those presented in the previous section. The cooperation percentage of each node represents, as always, the average number of rounds in which the node decides to cooperate, considering the last 250 rounds.

As shown in Figure 8.9, a higher percentage of users with positive values of α, produces higher cooperation levels when they play the extended versions of the PD, as we expected according to our framework. Then, we evaluated

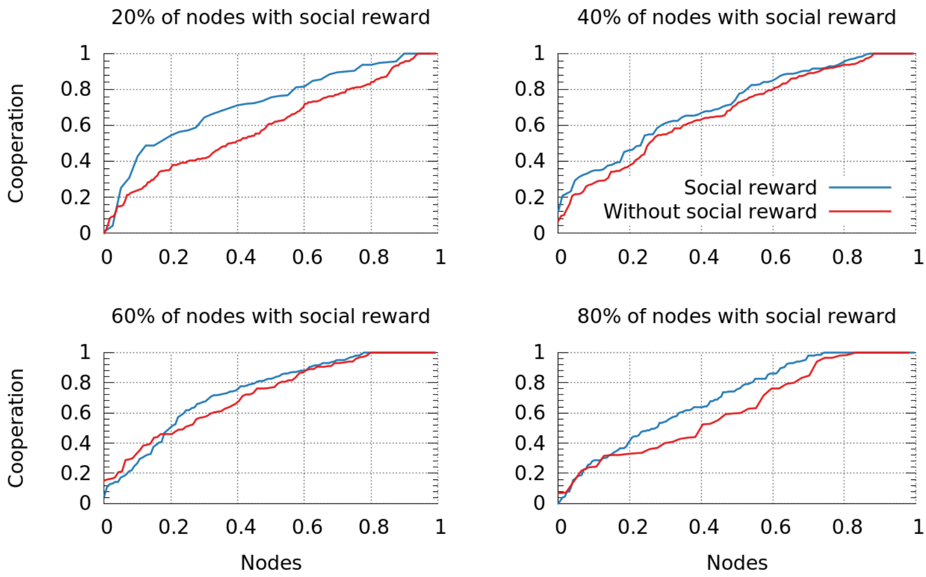


Figure 8.9: Users' cumulative cooperation distribution using a realistic distribution of social reward values.

the system using the same distribution of α, but with higher percentages of rewarded nodes. It was done by adapting the top percentile of messages (q).

We found that unlike the theoretical results — where the difference between the PD and the EPD strategies increases proportionally with the number of rewarded nodes —, when we applied a more realistic distribution, the improvement of the cooperation is almost unnoticeable if the size of both nodes groups remain similar. The most intuitive reason to explain such a phenomena is that more scaled social rewards could mask the difference between rewarded and no-rewarded nodes.

8.4 Generalizing the incentive mechanism

We have observed that it is possible to influence positively the cooperation ratio among users using the EPD strategy and a transfer function from some externality — in our case, the social score (α). To complete the incentive framework, in this section we present a generalization of our model, which includes the effort-based incentive mechanism discussed early on in this work (See Section 5), the new social score (See 8.3.6) and the transfer function.

Recall that we defined in our computational model the set of resources' slots available for a user i as a non empty set $R_i = \{R_i^1, R_i^2, R_i^3 \dots R_i^M\}$. Each job then, has an associated minimum cost in terms of the set of resource' slots W without which the job cannot be done. When a user i receives a request at time t from another user j, it shares a proportional number of slots to $P(W, t)$. In our generalized model, the percentage of resources is described by the Equation 8.1.

$$P(W,t) = \gamma\left(\frac{1}{M}\sum_{k=1}^{M}\frac{r_{ji}^k(t_{-1})}{R_j^k}\right) + \mu\left(\frac{1}{L}\sum_{l=1}^{L}\alpha_j(t_{-1})\right) \quad \textit{where } \gamma + \mu = 1 \tag{8.1}$$

The equation has two different components — the main activity and the different supporting activities — which are weighted by factors γ, μ.

$$\gamma, \mu = \{x \in \mathbb{R} \quad | \quad 0 \leq x \leq 1\} \tag{8.2}$$

The first component of Equation 8.1 denotes the sum of effort done by the user j sharing resources with node i in the previous round, for each type of resource k. The second component represents the sum of social efforts of user j on each of the L supporting activities. The parameter α, is the social score:

$$\alpha_i(t) = F(X : \beta_i(t) \to \mathbb{R}) \tag{8.3}$$

where X is a function provided by the system designer to transform each social position β into a real number, and the function F is its Probability Integral Transformation.

Notice that this generalization can be further expanded too, giving a different weight to each resource type k and to each supporting activity l. However, in practice the number of computational resources involved is not as large to require such expansion, and the social score can be simplified by using our extended framework for detecting roles and positions (See 7.4.3).

Chapter 9

Conclusions

This work makes several contributions to the design of incentive mechanisms for governing distributed infrastructures aimed to share computational resources unevenly distributed among nodes.

We firstly describe a regulation mechanism for increasing the cooperation opportunities of nodes with fewer resources and show how it indirectly improves the utilization of resources, while increases the overall amount of satisfied tasks in the system. The incentive is based on concepts from participatory economics, seeking to promote the cooperation willingness and effort rather than to just increase the absolute contribution of nodes — as other contributory-based mechanisms do.

The most important drawback of the mechanism is that it is not envy-free. Therefore, nodes with larger amount of resources might be reluctant to adopt the new policy, as some of them could get more resources from the system by switching to traditional contributory-based incentives. In this work we show how this situation can be avoided by reusing some of the ideas developed by Ostrom [6], and providing the community of users with the necessary tools to enforce and guarantee that nodes comply with the governing rules.

For a policy to being enforced by a community of practitioners, the existence of users able and willing to devote their time and effort must be guaranteed. We proposed a second incentive mechanism to encourage users' participation on supporting tasks by rewarding them with some benefits in the sharing environment. It required also developing a new framework for detecting roles and positions in multiplex graphs.

We additionally show how to encourage the cooperation among nodes with fewer resources in scarcity scenarios by providing and placing community-owned computational resources in key positions of the overlay network. We find evidences that topologies with some structural properties are more suitable than other for these tasks, and evaluated their performance in a collaborative scenario.

Limitations

The evaluation of the mechanisms presented in this work is limited by the computational model simulated and the simulator itself. It affects especially to the results related with the social behaviour of the supporting community.

In this work we were able to provide mechanisms for measuring the social participation and for influencing the collaborative process, but we were unable to test how these implementations would affect the social behaviour of a real collective. We can only assume that players are rational, and that as a consequence some of them would choose to participate in the supporting activities because the positive reward — in particular, selfish users who will try to avoid sharing their own resources.

Besides that, most of the scenarios modeled were assumed static. This was a conscious decision to simplify the model and easier the interpretation of the results. However, most components of cooperative applications (e.g., overlay network, users) have a dynamic behaviour. This observation is also relevant for modeling the relations between users in the participatory forums.

Implications

This thesis reflects the role of users' communities in the development of governing policies for distributed architectures, especially in encouraging and ruling the cooperative process of sharing resources. The integration of users' activities on computational common-pool resources opens new research perspectives to emerging computational architectures which, directly or indirectly, are adopting the common-pool resources model [34, 134]. The feasibility of each of the mechanisms proposed in this thesis, however, depends on the state of the community.

According to the life-cycle proposed by Iriberri et.al. [135], online communities are identified as *mature*, when "the need for a more explicit and formal organization with regulations, rewards for contributions, subgroups, and discussion of more or less specific topics is evident.". At this stage, the relation and interaction among members are long-lived, and most of the culture of communities of interest and communities of practice are evident.

Therefore, from this stage on if the community is attached to a cooperative application it would be ready to implement and enforce governing rules aimed to share common resources like, for example, the effort-based mechanisms described in Chapter 5. There is no need for enforcing the users' participation in the community or its supporting activities because even when the incorporation of new members might cause an adjustment of community roles — and modify its internal dynamics —, if the dialog is still present the excitement will continue nurturing the collective actions [136].

The social incentives (See Chapters 5 and 8) are needed, instead, during the *growth* stage, when the technological components are in place, and the roles are being or have already been established. Usually in this stage, lurkers — users consuming information from the online community without contributing — begin to appear, making more necessary the encouragement of volunteers and leaders to guarantee that the community arrives to the *mature* stage.

It is evident that before the community of users exists — or it is in its early stages: *inception* and *creation* —, the inventive policies based on effort measurements cannot be applied. Those mechanisms aimed for sharing physical resources are not effective because they need some externality to enforce them, while those measured as social interactions cannot be measured if the community does not exist. The lack of opportunities for cooperation on devices with fewer resources can only be fought by adding new resources to the common pool. This can be done because new users are participating of the main activity or because the new resources are contributed collectively (See Chapter 6).

This perspective highlights the community of supporters as the main reason to use each of the mechanisms presented in this work, despite that the community of users was initially a tool, and not a catalyst. This is important, as we have seen, in presence of different life-cycles between users — those participating of the main activity sharing resources — and supporters [P8].

Future directions

Regulating how cooperative users share resources is an old challenge. New research advancements in the areas of information systems and economics enable researchers to address more complex problems related with the main challenge. Two of the novel principles — participatory economics and collective governing policies — presented in this thesis are of particular interest from our point of view.

Using participatory economic principles for regulating computational pool resources has proved a good solution to build fairer incentives, but the underlaying consequences require further investigation. In this thesis we were able to support non envy-free incentives by using some externality — the community of users. However, most distributed applications might not have a community of practitioners in which delegate these tasks. Economic markets, however, have been proved stable solutions to implement dynamic regulations of computational resources. It would be of great interest to implement participatory economic principles in the bidding process of economic markets.

The user community as such has never been previously used as part of the solution in computer architectures. Previous works considered users just rational agents who intended to maximize their payoff. We have changed this perspective, and proposed a model where there is an ecosystem formed by supporters and regular users who interact with the distributed architecture — and the governing mechanisms — in different ways. Understand better how this interactions occur, when and what are the consequences for the resource sharing application are fundamental. In this work, for instance, we captured these interactions through the exchange of messages in online participatory forums, which will limit the scope of the information gathered, and might hide other forms of participation. Furthermore, we studied the effect of encouraging the participation of users in the supporting activities, by rewarding them in the main one which poses the next questions: Can we do it the other way around (encouraging the participation in the main activity by rewarding users in the supporting ones)? Can we apply both policies at the same time?

Acknowledgments

It has been a long, long journey; but also a great opportunity to meet and learn from very talented people without whom this thesis would not have been possible.

First and foremost, I would like to thank my advisor, Roc Meseguer. These past several years, he has been the most valuable support and — without noticing — a great source of inspiration. He always prompted a balanced combination of interesting challenges and good advice. In short, I would not arrived until here if it were not for him.

I am also very thankful to Leandro Navarro for his endless optimism. Additionally, his practicality has been proved a very useful tool to survive during all these years.

I also want to thank the Distributed Systems Group (DSG) and the partners of CONFINE and Clommunity projects for supporting my — sometimes crazy — research ideas. I want to thank all my current and former colleagues from the DSG, who created a wonderful working environment in the lab. Special thanks to Navaneeth Rameshan, for our philosophical discussions about many research topics; and to Ester López, who started this journey in the research group at the same time as me. During all these years she has been a model to follow. Thanks for your support, guys.

During all these years I had the luck to work in the Castelldefels School of Telecommunications and Aerospace Engineering (EETAC). I want to thank all the people to which I shared many hours in the Campus, but specially to Juan López, Esther Salamí and Pablo Royo who never have declined an offer to share a cup of coffee (to which I owe a big thanks).

I also owe many thanks to my colleagues from the Uppsala DataBase Laboratory — where I did a visit — for the interesting, fruitful and exciting six months. I would like to thank Matteo Magnani for being such a good host and for our endless conversations and board discussions. Even in the last stage of this thesis, he has been able to influence the way I understand scientific research.

I want to extend my thanks to Sobhan Badiozaman, Minpeng Zhu, Thanh Truong and Jonatan Samuelsson, who made my stay in Uppsala a great fun.

I would also thank to my officemate Virginia Grande, the most hard-worker person I have ever meet; to Åsa Cajander for her professional advice and Arnold Neville Pears for his (unfruitful) attempts to teach me Swedish.

Finally, this journey would have been even more longer and painful without the support and warm from my friends. Special thanks to Cristina Bosch, Javier López, Xavier Calvo, Daniel Rodriguez, Guillermo Enero and Guillem Cabrera for always being there. To my particular Quixote — Jorge I. Aguadero — for offering me caffeine when I needed so, and for believing so strongly in science. Definitely, the world needs more Jorges.

Most importantly, I wish to express a special gratitude to my family. To my parents, Milena and Joaquín for their continuous and unconditional love and support. My brother, whose optimism is contagious; and to my uncle who I have always admired. Finally, my biggest thanks goes to my grandparents which — despite being separated miles-away — I always felt with me.

Davide Vega D'Aurelio
Barcelona, 12th november 2015

Publications

P1. Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Understanding Collaboration in Volunteer Computing Systems. *Journal of Universal Computer Science*, 20(13):1738–1765, November 2014. DOI: 10.3217/jucs-020-13-1738.

P2. Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Effort-based incentives for resource sharing in collaborative volunteer applications. In *Proc. of IEEE 17th International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2013, pages 37–42. DOI: 10.1109/CSCWD.2013.6580936.

P3. Vega, D., Meseguer, R., Ochoa, S. F., Pino, J. A., Freitag, F., Medina, E., and Royo, D. Sharing hardware resources in heterogeneous computer-supported collaboration scenarios. *Integrated Computer-Aided Engineering*, 20(1):59–77, June 2013. DOI: 10.3233/ICA-120419.

P4. Vega, D., Medina, E., Meseguer, R., Royo, D., and Freitag, F. A Node Placement Heuristic to Encourage Resource Sharing in Mobile Computing. In *International Conference Computational Science and Its Applications*. Volume 6784. In LNCS. Springer Berlin Heidelberg, 2011, pages 540–555. ISBN: 978-3-642-21930-6. DOI: 10.1007/978-3-642-21931-3_42.

P5. Vega, D., Medina, E., Meseguer, R., Royo, D., Freitag, F., Ochoa, S. F., and Pino, J. A. Characterizing the effects of sharing hardware resources in mobile collaboration scenarios. In *Proc. of IEEE 15th International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2011, pages 465–472. ISBN: 978-1-4577-0387-4. DOI: 10.1109/CSCWD.2011.5960114.

P6. Vega, D., Magnani, M., Meseguer, R., and Freitag, F. Role and position detection in networks: reloaded. In *Proc. of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2015. Forthcoming.

P7. Vega, D., Magnani, M., Meseguer, R., and Freitag, F. Detection of roles and positions in multi-layer social networks. In *XXXV Sunbelt Conference of the International Network for Social Network Analysis (INSNA)*. 2015.

P8. Vega, D., Meseguer, R., and Freitag, F. Analysis of the Social Effort in Multiplex Participatory Networks. In *Revised Selected Papers of 11th International Conference Economics of Grids, Clouds, Systems, and Services*. Volume 8914. In LNCS. Springer Berlin Heidelberg, 2014, pages 67–79. ISBN: 978-3-319-14608-9. DOI: 10.1007/978-3-319-14609-6_5.

P9. Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Motivating the non-technical participation in technical communities. In *Proc. of IEEE 18th International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2015. DOI: 10.1109/CSCWD.2015.7230968.

Other publications

R1. Millán, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B., and Blondia, C. Time Series Analysis to Predict Link Quality of Wireless Community Networks. *Computer Networks*, 2015. DOI: 10.1016/j.comnet.2015.07.021.

R2. Vega, D., Baig, R., Cerdà-Alabern, L., Medina, E., Meseguer, R., and Navarro, L. A Technological Overview of the Guifi.net Community Network. *Computer Networks*, 2015. DOI: 10.1016/j.comnet.2015.09.023.

R3. Millán, P., Molina, C., Medina, E., Vega, D., Meseguer, R., Braem, B., and Blondia, C. Tracking and predicting link quality in wireless community networks. In *Proc. of IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2014, pages 239–244. DOI: 10.1109/WiMOB.2014.6962177.

R4. Vega, D., Cerdà-Alabern, L., Navarro, L., and Meseguer, R. Topology patterns of a community network: Guifi.net. In *Proc. of IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE Computer Society, 2012, pages 612–619. DOI: 10.1109/WiMOB.2012.6379139.

R5. Vega, D., Meseguer, R., Cabrera, G., and Marquès, J. M. Exploring local service allocation in Community Networks. In *Proc. of IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*. 2014, pages 273–280. DOI: 10.1109/WiMOB.2014.6962182.

R6. Selimi, M., Florit, J. L., Vega, D., Meseguer, R., López, E., Khan, A. M., Neumann, A., et al. Cloud-Based Extension for Community-Lab. In *IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*. IEEE, 2014, pages 502–505. ISBN: 978-1-4799-5610-4. DOI: 10.1109/MASCOTS.2014.73. Demonstration.

R7. Garcia, P. E., Baig, R., Balaguer, I. V. i, Neumann, A., Aymerich, M., López, E., Vega, D., et al. Community home gateways for P2P clouds. In *Proc. of IEEE 13th International Conference on Peer-to-Peer Computing*. IEEE, 2013, pages 1–2. ISBN: 978-1-4799-0521-8. DOI: `10.1109/P2P.2013.6688732`. Poster.

R8. Aymerich, M., Baig, R., Garcia, P. E., Balaguer, I. V. i, Neumann, A., Vega, D., López, E., et al. Deploying applications with Community-Lab in wireless community networks. In *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"*. IEEE Computer Society, 2013, pages 1–3. ISBN: 978-1-4673-5827-9. DOI: `10.1109/WoWMoM.2013.6583362`. Demonstration.

R9. Vega, D., Meseguer, R., Freitag, F., and Ochoa, S. F. Motivating the non-technical participation in technical communities. In *XXIII Jornadas de Concurrencia y Sistemas Distribuidos"*. 2015.

R10. Vega, D., Meseguer, R., and Navarro, L. Analysis of the web proxy service of a community network: Guifi.net. In *XXII Jornadas de Concurrencia y Sistemas Distribuidos"*. 2014, pages 47–64.

R11. Vega, D., Meseguer, R., Freitag, F., and Navarro, L. Incentivos basados en el esfuerzo para compartir recursos en aplicaciones colaborativas. In *XXI Jornadas de Concurrencia y Sistemas Distribuidos"*. 2013.

R12. Vega, D., Meseguer, R., and Freitag, F. Diseño e implementación de un simulador para explorar la cooperación en entornos distribuidos. In *XIX Jornadas de Concurrencia y Sistemas Distribuidos"*. 2011, pages 311–326. ISBN: 84-96737-99-0.

Bibliography

1. Sarmenta, L. F. and Hirano, S. Bayanihan: building and studying web-based volunteer computing systems using Java. *Future Generation Computer Systems*, 15(5–6):675–686, December 1999. ISSN: 0167-739X. DOI: 10.1016/S0167-739X(99)00018-7.

2. Anderson, D. P. BOINC: a system for public-resource computing and storage. In *Proc. of IEEE/ACM 5th International Workshop on Grid Computing*. ACM, 2004. ISBN: 0-7695-2256-4. DOI: 10.1109/GRID.2004.14.

3. Iglesias, D. L. A Middleware for Service Deployment in Contributory Computing Systems. PhD thesis. Universitat Oberta de Catalunya. Internet Interdisciplinary Institute (IN3), 2011.

4. Johnson, R. T. and Johnson, D. W. *Cooperative Learning: Two Heads Learn Better Than One. Transforming Education.* Obtido em, 1988, page 34.

5. Hardin, G. The Tragedy of the Commons. *Science*, 162(3859):1243–1248, December 1968. DOI: 10.1126/science.162.3859.1243.

6. Ostrom, E. *Governing the commons: The evolution of institutions for collective action.* Cambridge university press, 1990.

7. Albert, M. and Hahnel, R. *Looking forward: Participatory economics for the twenty first century.* South End Press, 1991.

8. Nowak, M. A. Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563, December 2006. DOI: 10.1126/science.1133755.

9. Fan, B., Chiu, D.-M., and Lui, J. C. The delicate tradeoffs in bittorrent-like file sharing protocol design. In *Proc. of 14th IEEE International Conference on Network Protocols*. IEEE, 2006, pages 239–248. ISBN: 1-4244-0593-9. DOI: 10.1109/ICNP.2006.320217.

10. Cassar, A. Coordination and cooperation in local, random and small world networks: Experimental evidence. *Games and Economic Behavior*, 58(2):209–230, September 2007. ISSN: 0899-8256. DOI: 10.1016/j.geb.2006.03.008.

11. Lozano, S., Arenas, A., and Sánchez, A. Mesoscopic Structure Conditions the Emergence of Cooperation on Social Networks. *PLoS ONE*, 3(2):189–192, April 2008. DOI: 10.1371/journal.pone.0001892.

12. Santos, F. C., Rodrigues, J. F., and Pacheco, J. M. Graph topology plays a determinant role in the evolution of cooperation. *Proceedings of the Royal Society of London B: Biological Sciences*, 273(1582):51–55, January 2006. ISSN: 0962-8452. DOI: 10.1098/rspb.2005.3272.

13. Lakhani, K. R. and Hippel, E. V. How open source software works:"free" user-to-user assistance. *Research policy*, 32(6):923–943, June 2003. DOI: 10.1016/S0048-7333(02)00095-1.

14. Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. Finding High-quality Content in Social Media. In *Proc. of ACM 1st International Conference on Web Search and Data Mining*. ACM, 2012, pages 183–194. ISBN: 978-1-59593-927-2. DOI: 10.1145/1341531.1341557.

15. Wasserman, S. *Social network analysis: Methods and applications*. Volume 8. Cambridge University Press, 1994.

16. Žiberna, A. Generalized blockmodeling of valued networks. *Social networks*, 29(1):105–126, September 2007. DOI: 10.1016/j.socnet.2006.04.002.

17. Foster, I. and Kesselman, C. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., 1999. ISBN: 1-55860-475-8.

18. Foster, I., Yong, Z., Raicu, I., and Shiyong, L. Cloud Computing and Grid Computing 360-Degree Compared. In *Proc. of Grid Computing Environments Workshop*. 2008, pages 1–10. DOI: 10.1109/GCE.2008.4738445.

19. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., et al. A View of Cloud Computing. *Commun. ACM*, 53(4):50–58, April 2010. ISSN: 0001-0782. DOI: 10.1145/1721654.1721672.

20. Pouwelse, J., Garbacki, P., Epema, D., and Sips, H. The Bittorrent P2P File-Sharing System: Measurements and Analysis. In *Proc. of 4th International Workshop in Peer-to-Peer systems*. Volume 3640. In Lecture Notes in Computer Science. Springer, 2009, pages 205–216. ISBN: 978-3-540-29068-1. DOI: 10.1007/11558989_19.

21. Weber, S. *The success of open source*. Volume 897. Cambridge University Press, 2004. ISBN: 9780674018587.

22. Wenger, E. *Communities of practice: Learning, meaning, and identity*. Cambridge University Press, 1999. ISBN: 9780521663632.

23. Wenger, E. Communities of Practice and Social Learning Systems: the Career of a Concept. In Blackmore, C., editor, *Social Learning Systems and Communities of Practice*, pages 179–198. Springer London, 2010. ISBN: 978-1-84996-132-5. DOI: 10.1007/978-1-84996-133-2_11.

24. Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proc. of ACM 15th Conference on Computer Supported Cooperative Work*. ACM, 2012, pages 1277–1286. ISBN: 978-1-4503-1086-4. DOI: 10.1145/2145204.2145396.

25. Braem, B., Blondia, C., Barz, C., Rogge, H., Freitag, F., Navarro, L., Bonicioli, J., et al. A Case for Research with and on Community Networks. *ACM SIGCOMM Computer Communication Review*, 43(3):68–73, June 2013. ISSN: 0146-4833. DOI: 10.1145/2500098.2500108.

26. Guifi.net: Open, Free and Neutral Network Internet for everybody.
http://guifi.net/en.

27. FunkFeuer free net.
http://www.funkfeuer.at.

28. Avonts, J., Braem, B., and Blondia, C. A questionnaire based examination of community networks. In *Proc. of 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2013, pages 8–15. DOI: 10.1109/WiMOB.2013.6673333.

29. Marinos, A. and Briscoe, G. Community Cloud Computing. In *Proc. of 1st Conference in Cloud Computing*. In Lecture Notes in Computer Science. Springer, 2009, pages 472–484. ISBN: 978-3-642-10664-4. DOI: 10.1007/978-3-642-10665-1_43.

30. Baig, R., Roca, R., Freitag, F., and Navarro, L. guifi.net, a crowdsourced network infrastructure held in common. *Computer Networks*, 2015. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2015.07.009.

31. Cunsolo, V., Distefano, S., Puliafito, A., and Scarpa, M. Volunteer Computing and Desktop Cloud: The Cloud@Home Paradigm. In *Proc. of 8th International Symposium on Network Computing and Applications*. IEEE, 2009, pages 134–139. ISBN: 978-0-7695-3698-9. DOI: 10.1109/NCA.2009.41.

32. Guarracino, M., Vivien, F., Träff, J.-L., Cannatoro, M., Danelutto, M., Hast, A., Perla, F., et al. The Cloud@Home Project: Towards a New Enhanced Computing Paradigm. In *Proc. of 16th International Euro-Par Conference*. Volume 6586. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pages 555–562. ISBN: 978-3-642-21877-4. DOI: 10.1007/978-3-642-21878-1_68.

33. Cunsolo, V., Distefano, S., Puliafito, A., and Scarpa, M. Cloud@Home: Bridging the Gap between Volunteer and Cloud Computing. In *Proc. of 5th International Conference on Intelligent Computing*. Volume 5754. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pages 423–432. ISBN: 978-3-642-04069-6. DOI: 10.1007/978-3-642-04070-2_48.

34. Marinos, A. and Briscoe, G. Community Cloud Computing. In *Proc. of 1st International Conference on Cloud Computing*. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pages 472–484. ISBN: 978-3-642-10664-4. DOI: 10.1007/978-3-642-10665-1_43.

35. Briscoe, G. and Marinos, A. Digital ecosystems in the clouds: Towards community cloud computing. In *Proc. of IEEE 3th International Conference on Digital Ecosystems and Technologies*. IEEE, 2009, pages 103–108. ISBN: 978-0-7695-3698-9. DOI: 10.1109/DEST.2009.5276725.

36. Khan, A., Navarro, L., Sharifi, L., and Veiga, L. Clouds of small things: Provisioning infrastructure-as-a-service from within community networks. In *Proc. of IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE Computer Society, 2013, pages 16–21. DOI: 10.1109/WiMOB.2013.6673334.

37. Rapoport, A. and M. Chammah, lbert. *Prisoner's dilemma: A study in conflict and cooperation*. Volume 165. University of Michigan press, 1965.

38. Osborne, M. J. *An introduction to game theory*. Volume 3(3). Oxford University Press New York, 2004.

39. Rahman, R., Meulpolder, M., Hales, D., Pouwelse, J., Epema, D., and Sips, H. Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives. In *Proc. of IEEE International Conference on Communications*. IEEE, 2010, pages 1–5. DOI: 10.1109/ICC.2010.5502544.

40. Axelrod, R. Effective Choice in the Prisoner's Dilemma. *The Journal of Conflict Resolution*, 24(1):3–25, March 1980. ISSN: 0022–0027.

41. Qiu, D., Srikant, R., and Lui, J. C. Modeling and Performance Analysis of BitTorrent-like Peer-to-peer Networks. In *Proc. of 14th ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, 2004, pages 367–378. ISBN: 1-58113-862-8. DOI: 10.1145/1015467.1015508.

42. Levin, D., LaCurts, K., and Neil Spring, B. B. Bittorrent is an Auction: Analyzing and Improving Bittorrent's Incentives. *SIGCOMM Computer Communications. Review*, 38(4):243–254, August 2008. DOI: 10.1145/1402946.1402987.

43. Robert Axelrod and, W. D. H. The evolution of cooperation. *Science*, 211(4489):1390–1396, March 1981. DOI: 10.1126/science.7466396.

44. Jun, S. and Ahamad, M. Incentives in BitTorrent Induce Free Riding. In *Proc. of ACM SIGCOM Workshop on Economics of Peer-to-Peer Systems*. ACM, 2005, pages 116–121. DOI: 10.1145/1080192.1080199.

45. Iamnitchi, A., Ripeanu, M., and Foster, I. Small-world file-sharing communities. In *Proc. of 23th Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, 2004, pages 952–963. DOI: 10.1109/INFCOM.2004.1356982.

46. Scott, J. P. and Carrington, P. J. *Social Network Analysis*. Sage Publications Ltd., 2013. ISBN: 9781446209035.

47. Erdős, P. and Rényi, A. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

48. Waxman, B. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988. ISSN: 0733-8716. DOI: 10.1109/49.12889.

49. Watts, D. J. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998. DOI: 10.1038/30918.

50. Barabási, A.-L. and Albert, R. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 2015. ISSN: 1095-9203. DOI: 10.1126/science.286.5439.509.

51. Breiger, R. L. and Pattison, P. E. Cumulated social roles: The duality of persons and their algebras. *Social Networks*, 8(13):215–256, 1986. ISSN: 0378-8733. DOI: 10.1016/0378-8733(86)90006-7.

52. Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. Multilayer networks. *Journal of Complex Networks*, July 2014. DOI: 10.1093/comnet/cnu016.

53. Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77, December 2007. DOI: 10.2753/MIS0742-1222240302.

54. Otte, E. and Rousseau, R. Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441–453, December 2002. DOI: 10.1177/016555150202800601.

55. Juhola, M. and Siermala, V. A Scatter Method for Data and Variable Importance Evaluation. *Integrated Computer-Aided Engineering*, 19(2):137–149, April 2013. ISSN: 1069-2509. DOI: 10.3233/ICA-2011-0385.

56. Guyon, I. and Elisseeff, A. An Introduction to Variable and Feature Selectionn. *Journal of Machine Learning Research*, 3:1157–1182, January 2013. ISSN: 1532-4435.

57. Hall, M. A. Correlation-based Feature Selection for Machine Learning. PhD thesis. University of Waikato, 1999.

58. Kononenko, I. Estimating attributes: Analysis and extensions of RELIEF. In *Proc. of European Conference on Machine Learning.* In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1994, pages 171–182. ISBN: 978-3-540-57868-0. DOI: 10.1007/3-540-57868-4_57.

59. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, January 2009. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278.

60. Palla, G., Derenyi, I., Farkas, I., and Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, January 2005. DOI: 10.1038/nature03607.

61. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):100–108, October 2008.

62. Kleinberg, J. Hubs, Authorities, and Communities. *ACM Computer Survery*, 31(4):29–53, 1999. ISSN: 0360-0300. DOI: 10.1145/345966.345982.

63. Doreian, P. Equivalence in a social network. *Journal of mathematical sociology*, 13(3):243–281, 1988.

64. Lorrain, F. and White, H. C. Structural equivalence of individuals in social networks. *Journal of mathematical sociology*, 1(1):49–80, 1971.

65. Generalized blockmodeling. SFI/SAS Workshop. The dynamics of groups and institutions.
http://vlado.fmf.uni-lj.si/pub/networks/doc/seminar/SFI-SAS04bm.pdf.

66. Campbell, A. T., Eisenman, S. B., Lane, N. D., Miluzzo, E., and Peterson, R. A. People-centric Urban Sensing. In *Proc. of 2nd Annual International Workshop on Wireless Internet.* ACM, 2006. ISBN: 1-59593-510-X. DOI: 10.1145/1234161.1234179.

67. Ganti, R., Ye, F., and Lei, H. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, November 2011. ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.6069707.

68. Ochoa, S. F., Bravo, G., Pino, J. A., and Rodríguez-Covili, J. Coordinating Loosely-Coupled Work in Construction Inspection Activities. *Group Decision and Negotiation*, 20(1):39–56, 2011. ISSN: 0926-2644. DOI: 10.1007/s10726-010-9203-9.

69. Monares, Á., Ochoa, S. F., Pino, J. A., Herskovic, V., Rodriguez-Covili, J., and Neyem, A. Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field. *Group Decision and Negotiation*, 38(2):1255–1267, February 2011. ISSN: 0957-4174. DOI: `10.1016/j.eswa.2010.05.018`.

70. Cankar, M., Artač, M., Šterk, M., Lotrič, U., and Slivnik, B. Co-Allocation with Collective Requests in Grid Systems. *Journal of Universal Computer Science*, 19(3):282–300, February 2013. DOI: `10.3217/jucs-019-03-0282`.

71. Bittencourt, L. F., Miyazawa, F. K., and Vignatti, A. L. Distributed Load Balancing Algorithms for Heterogeneous Players in Asynchronous Networks. *Journal of Universal Computer Science*, 18(20):282–300, December 2012. DOI: `10.3217/jucs-019-03-0282`.

72. Chandra, A. and Weissman, J. Nebulas: Using Distributed Voluntary Resources to Build Clouds. In *Proc. of IEEE/ACM 5th International Workshop on Grid Computing*. USENIX Association, 2009.

73. Kannan, S., Gavrilovska, A., and Schwan, K. Cloud4Home – Enhancing Data Services with @Home Clouds. In *Proc. of IEEE 31st International Conference on Distributed Computing Systems*. IEEE, 2011, pages 539–548. DOI: `10.1109/ICDCS.2011.74`.

74. Lazaro, D., Marques, J. M., and Jorba, J. Towards an Architecture for Service Deployment in Contributory Communities. *International Journal of Grid and Utility Computing*, 1(3):227–238, August 2009. DOI: `10.1504/IJGUC.2009.027650`.

75. Chmaj, G. and Walkowiak, K. Decision Strategies for a P2P Computing System. *Journal of Universal Computer Science*, 18(5):599–622, March 2012. DOI: `10.3217/jucs-018-05-0599`.

76. Chard, K., Bubendorfer, K., Caton, S., and Rana, O. Social Cloud Computing: A Vision for Socially Motivated Resource Sharing. *IEEE Transactions on Services Computing*, 5(4):551–563, April 2012. ISSN: 1939-1374. DOI: `10.1109/TSC.2011.39`.

77. Rand, D. G., Dreber, A., Ellingsen, T., Fudenberg, D., and Nowak, M. A. Positive Interactions Promote Public Cooperation. *Science*, 325(5945):1272–1275, 2009. DOI: `10.1126/science.1177418`.

78. Feldman, M. and Lai, J. Mechanisms and Impossibilities for Truthful, Envy-Free Allocations. In *Proc. of 5th International Symposium in Algorithmic Game Theory*. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pages 120–131. ISBN: 978-3-642-33995-0. DOI: `10.1007/978-3-642-33996-7_11`.

79. Santos, A., Fernández, A., and López, L. Quid Pro Quo: A Mechanism for Fair Collaboration in Networked Systems. *PLoS ONE*, 8(9), September 2013. DOI: 10.1371/journal.pone.0066575.

80. The BitTorrent Protocol Specification. http://www.BitTorrent.org/beps/bep_0003.html.

81. Bharambe, A., Herley, C., and Padmanabhan, V. Analyzing and Improving a BitTorrent Networks Performance Mechanisms. In *Proc. of IEEE 25th International Conference on Computer Communications*. IEEE, 2006, pages 1–12. DOI: 10.1109/INFOCOM.2006.328.

82. Piatek, M., Isdal, T., Anderson, T., Krishnamurthy, A., and Venkataramani, A. Do Incentives Build Robustness in Bit Torrent? In *Proc. of USENIX 4th Conference on Networked Systems Design and Implementation*. USENIX Association, 2006, pages 1–1.

83. Lee, B.-D. and Weissman, J. Fast download but eternal seeding: The reward and punishment of Sharing Ratio Enforcement. In *Proc. of IEEE 11th International Conference on Peer-to-Peer Computing*. IEEE, 2011, pages 280–289. DOI: 10.1109/P2P.2011.6038746.

84. Legout, A., Liogkas, N., Kohler, E., and Zhang, L. Clustering and Sharing Incentives in BitTorrent Systems. In *Proc. of ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, 2007, pages 280–289. ISBN: 978-1-59593-639-4. DOI: 10.1145/1254882.1254919.

85. Zhang, C., Dhungel, P., Wu, D., Liu, Z., and Ross, K. BitTorrent Darknets. In *Proc. of IEEE 29th International Conference on Computer Communications*. IEEE, 2010, pages 1–9. DOI: 10.1109/INFCOM.2010.5461962.

86. Chen, X., Jiang, Y., and Chu, X. Measurements, Analysis and Modeling of Private Trackers. In *Proc. of IEEE 10th International Conference on Peer-to-Peer Computing*. IEEE, 2010, pages 1–10. DOI: 10.1109/P2P.2010.5569968.

87. Jia, A., D'Acunto, L., Meulpolder, M., Pouwelse, J., and Epema, D. BitTorrent's dilemma: Enhancing reciprocity or reducing inequity. In *Proc. of IEEE Consumer Communications and Networking Conference*. IEEE, 2011, pages 705–709. DOI: 10.1109/CCNC.2011.5766581.

88. Willer, D. *Network exchange theory*. Greenwood Publishing Group, 1999.

89. Barabási, A.-L. and Frangos, J. *Linked: the new science of networks science of networks*. Basic Books, 2014.

90. Lilja, D. J. *Measuring computer performance: a practitioner's guide*. Cambridge University Press, 2000.

91. Pinelle, D. and Gutwin, C. Loose Coupling and Healthcare Organizations: Deployment Strategies for Groupware. *Computer Supported Cooperative Work (CSCW)*, 15(5–6):537–572, December 2006. ISSN: 0925-9724. DOI: 10.1007/s10606-006-9031-2.

92. Mejía, D. A., Favela, J., and Morán, A. L. Preserving Interaction Threads through the Use of Smartphones in Hospitals. In *Proc. of 15th International Conference on Collaboration and Technology*. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pages 17–31. ISBN: 978-3-642-04215-7. DOI: 10.1007/978-3-642-04216-4_2.

93. Guerrero, L. A., Ochoa, S. F., Pinto, J. A., and Collazos, C. A. Selecting Computing Devices to Support Mobile Collaboration. *Group Decision and Negotiation*, 15(3):243–271, June 2006. ISSN: 0926-2644. DOI: 10.1007/s10726-006-9020-3.

94. Rodrigues, R. and Druschel, P. Peer-to-peer Systems. *Commun. ACM*, 53(10):72–82, October 2010. ISSN: 0001-0782. DOI: 10.1145/1831407.1831427.

95. Chevaleyre, Y., Endriss, U., Lang, J., and Maudet, N. A Short Introduction to Computational Social Choice. In *Proc. of 33rd International Conference on Current Trends in Theory and Practice of Computer Science*. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pages 51–69. ISBN: 978-3-540-69506-6. DOI: 10.1007/978-3-540-69507-3_4.

96. Cornuéjols, G., Nemhauser, G. L., and Wolsey, L. A. The uncapacitated facility location problem. *Discrete Location Theory*, 1990.

97. Coppens, J., Wauters, T., Turck, F. D., Dhoedt, B., and Demeester, P. Evaluation of replica placement and retrieval algorithms in self-organizing CDNs. In *Proc. of IFIP/IEEE International Workshop on Self-Managed Systems and Services*. ACM, 2012, pages 13–16.

98. Karlsson, M. and Mahalingam, M. Do We Need Replica Placement Algorithms in Content Delivery Networks. In *Proc. of 7th International Workshop on Web Content Caching and Distribution*, 2000, pages 117–128.

99. Tang, X., Chi, H., and Chanson, S. Optimal Replica Placement under TTL-Based Consistency. *IEEE Transactions on Parallel and Distributed Systems*, 18(3):351–363, 2007. ISSN: 1045-9219. DOI: 10.1109/TPDS.2007.47.

100. Lee, B.-D. and Weissman, J. Dynamic replica management in the service grid. In *Proc. of IEEE 10th International Symposium on High Performance Distributed Computing*. IEEE, 2001, pages 433–434. DOI: 10.1109/HPDC.2001.945213.

101. Graupner, S., Andrzejak, A., Kotov, V., and Trinks, H. Adaptive Service Placement Algorithms for Autonomous Service Networks. In *Engineering Self-Organising Systems*. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pages 280–297. ISBN: 978-3-540-26180-3. DOI: `10.1007/11494676_19`.

102. Liu, K. Y., Lui, J. C., and Zhang, Z.-L. Distributed Algorithm for Service Replication in Service Overlay Network. In *Proc. of 3rd International IFIP-TC6 Networking Conference*. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pages 1156–1167. ISBN: 978-3-540-21959-0. DOI: `10.1007/978-3-540-24693-0_95`.

103. Herrmann, K. Self-organized Service Placement in Ambient Intelligence Environments. *ACM Transactions on Autonomous and Adaptive Systems*, 5(2):1–39, 2010. ISSN: 1556-4665. DOI: `10.1145/1740600.1740602`.

104. Santos, R., Orozco, J., and Ochoa, S. F. A Real-time Analysis Approach in Opportunistic Networks. *ACM SIGBED Review*, 8(3):29–53, 2011. ISSN: 1551-3688. DOI: `10.1145/2038617.2038626`.

105. Pawlikowski, K., Jeong, H.-D., and Lee, J.-S. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, 40(1):132–139, October 2010. ISSN: 0163-6804. DOI: `10.1109/35.978060`.

106. Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, September 1933. DOI: `10.1037/h0071325`.

107. Kleinberg, J. The Small-world Phenomenon: An Algorithmic Perspective. In *Proc. of 32th Annual ACM Symposium on Theory of Computing*. 2000, pages 163–170. ISBN: 1-58113-184-4. DOI: `10.1145/335305.335325`.

108. Wang, Y. and Nakao, A. On Socially-inspired Cooperative and Efficient Overlay Network Evolution Based on Group Selection Pattern. In *Proc. of 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Sytems*. ISBN: 978-963-9799-35-6.

109. Gini, C. *On the Measure of Concentration with Espacial Reference to Income and Wealth*. Cowles Commission, 1936.

110. Miller, K. T., Griffiths, T. L., and Jordan, M. I. Nonparametric Latent Feature Models for Link Prediction. In *Proc. of 26th Annual Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2009, pages 1276–1284.

111. Doreian, P., Batagelj, V., and Ferligoj, A. Generalized blockmodeling of two-mode network data. *Social Networks*, 26(1):29–53, 2004. ISSN: 0378-8733. DOI: `10.1016/j.socnet.2004.01.002`.

112. Patrick Doreian and, V. B. and Ferligoj, A. *Generalized Blockmodeling*. Cambridge University Press, 2005. ISBN: 9780521840859.

113. Brusco, M., Doreian, P., Steinley, D., and Satornino, C. B. Multiobjective Blockmodeling for Social Network Analysis. *Psychometrika*, 78(3):498–525, 2013. ISSN: 0033-3123. DOI: 10.1007/s11336-012-9313-1.

114. Rossi, R. A. and Ahmed, N. K. Role Discovery in Networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1112–1131, April 2015. ISSN: 0033-3123. DOI: 10.1109/TKDE.2014.2349913.

115. Welser, H. T., Cosley, D., Kossinets, G., Lin, A., Dokshin, F., Gay, G., and Smith, M. Finding Social Roles in Wikipedia. In *Proc. of the iConference*. ACN, 2011, pages 122–129. ISBN: 978-1-4503-0121-3. DOI: 10.1145/1940761.1940778.

116. Buntain, C. and Golbeck, J. Identifying Social Roles in Reddit Using Network Structure. In *Proc. of the 23rd International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2014, pages 615–620. ISBN: 978-1-4503-2745-9. DOI: 10.1145/2567948.2579231.

117. White, D. R. and Reitz, K. P. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2):193–234, 1983. ISSN: 0378-8733. DOI: 10.1016/0378-8733(83)90025-4.

118. Borgatti, S. P. and Everett, M. G. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361–376, 1993. ISSN: 0378-8733. DOI: 10.1016/0378-8733(93)90012-A.

119. Murtagh, F. and Legendre, P. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, 2014. ISSN: 0176-4268. DOI: 10.1007/s00357-014-9161-z.

120. Žiberna, A. R blockmodeling-package. https://cran.r-project.org/web/packages/blockmodeling/blockmodeling.pdf.

121. Rossi, L. and Magnani, M. Towards effective visual analytics on multiplex and multilayer networks. *Chaos, Solitons and Fractals*, 72:68–75, 2015. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2014.12.022.

122. Pfitzmann, A. and Köhntopp, M. International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability. In *Proc. of International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*. Springer-Verlag New York, 2001, pages 1–9. ISBN: 3-540-41724-9.

123. Sowe, S. K., Stamelos, I., and Angelis, L. Free-riding and Whitewashing in Peer-to-peer Systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1010–1019, September 2006. ISSN: 0733-8716. DOI: 10.1109/JSAC.2006.872882.

124. Bird, C., Gourley, A., Devanbu, P., Gertz, M., and Swaminathan, A. Keep Your Friends Close and Your Facebook Friends Closer: A Multiplex Network Approach to the Analysis of Offline and Online Social Ties. In *Proc. of 3rd International Workshop on Mining Software Repositories*. ACM, 2006, pages 137–143. ISBN: 1-59593-397-2. DOI: 10.1145/1137983.1138016.

125. Nowak, M. A., Derenyi, I., and May, R. M. Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829, October 1992. DOI: 10.1038/359826a0.

126. Nash, J. F. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, January 1950.

127. Grujić, J., Fosco, C., Araujo, L., Cuesta, J. A., and Sánchez, A. Social Experiments in the Mesoscale: Humans Playing a Spatial Prisoner's Dilemma. *PLoS ONE*, 5(11), November 2010. DOI: 10.1371/journal.pone.0013749.

128. Fernández-García, J., Suchecki, K., Ramasco, J. J., Cuesta, M. S. M. A., and Eguíluz, V. M. Social Experiments in the Mesoscale: Humans Playing a Spatial Prisoner's Dilemma. *Physics Review Letters*, 112(15):1587–1589, April 2014. DOI: 10.1103/PhysRevLett.112.158701.

129. Grujić, J., Gracia-Lázaro, C., Milinski, M., Semmann, D., Traulsen, A., Cuesta, J. A., Moreno, Y., et al. A comparative analysis of spatial Prisoner's Dilemma experiments: Conditional cooperation and payoff irrelevance. *Scientific Reports*, 4(4615), October 2014. DOI: 10.1038/srep04615.

130. Vilone, D., Ramasco, J. J., Sánchez, A., and San-Miguel, M. Social Experiments in the Mesoscale: Humans Playing a Spatial Prisoner's Dilemma. *Physics Review Letters*, 90(2), August 2014. DOI: 10.1103/PhysRevE.90.022810.

131. Sefton, M., Shupp, R., and Walker, J. M. The effect of rewards and sanctions in provision of public goods. *Economic Inquiry*, 45(4):671–690, August 2007. ISSN: 1465-7295. DOI: 10.1111/j.1465-7295.2007.00051.x.

132. Sowe, S. K., Stamelos, I., and Angelis, L. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems and Software*, 81(3):431–446, March 2008. ISSN: 0164-1212. DOI: 10.1016/j.jss.2007.03.086.

133. Hristova, D., Musolesi, M., and Mascolo, C. Keep Your Friends Close and Your Facebook Friends Closer: A Multiplex Network Approach to the Analysis of Offline and Online Social Ties. In *Proc. of 8th International Conference on Weblogs and Social Media*. AAAI Press, 2014. ISBN: 978-1-57735-659-2.

134. Harrer, A. and Schmidt, A. Fog Computing and Its Role in the Internet of Things. In *Proc. of 1st MCC Workshop on Mobile Cloud Computing*. ACM, 2012, pages 13–16. ISBN: 978-1-4503-1519-7. DOI: 10.1145/2342509.2342513.

135. Iriberri, A. and Leroy, G. A Life-cycle Perspective on Online Community Success. *ACM Computing Surveys*, 41(2):509–512, 2009. ISSN: 0360-0300. DOI: 10.1145/1459352.1459356.

136. Wenger, E., McDermott, R. A., and Snyder, W. *Cultivating communities of practice: A guide to managing knowledge*. Harvard Business Press, 2002.